

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Миколенко Едуарду Владиславовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методів видалення шуму з відеоданих

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 7 січня 2024 р.3. Вихідні дані до роботи теоретичні відомості про типи шуму у відеорядах, популярні методи їх усування, реалізація методу видалення шуму.

4. Перелік питань, що потрібно опрацювати в роботі _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
зображення

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	10.11.23-12.11.23	
3	Аналіз літератури з досліджуваної проблеми	12.11.23-13.11.23	
4	Аналіз технічних засобів	14.11.23-17.11.23	
5	Розробка методів	18.12.23-22.12.23	
6	Програмна реалізація	23.12.23-27.12.23	
7	Оформлення пояснювальної записки	30.12.23-03.12.23	
8	Перевірка на плагіат	05.01.2023	
9	Рецензування	06.01.2023	
10	Підготовка презентації та доповіді	07.01.2023	
11	Занесення роботи в електронний архів	16.01.2024	
12	Попередній захист кваліфікаційної роботи	16.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

ст. викл. Путятіна О.Є.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 63 с., 7 рис., 31 джерело.

З впровадженням цифрових технологій у сферу відеозапису та обробки відеоданих, актуальною стає проблема шуму, що негативно впливає на якість відео.

Методики обробки відеосигналу, такі як фільтрація, адаптивні фільтри, і використання машинного та глибокого навчання, включаючи нейронні мережі, є основними напрямками досліджень. Ці методики дозволяють не тільки виявляти та видаляти шум, але й класифікувати його типи для більш ефективного очищення відео.

Оскільки відео є одним з основних засобів зберігання та передачі інформації, покращення якості відеоматеріалів має значний потенціал для збільшення їхньої цінності в різних областях. Ці дослідження вимагають глибокого знання в сферах сигнальної обробки, комп'ютерного зору, машинного навчання та розуміння специфіки відеоданих.

With the introduction of digital technologies in the field of video recording and video data processing, the problem of noise, which negatively affects video quality, becomes relevant.

Video signal processing techniques such as filtering, adaptive filters, and the use of machine and deep learning, including neural networks, are the main areas of research. These techniques allow not only to detect and remove noise, but also to classify its types for more effective video cleaning.

Since video is one of the main means of storing and transmitting information, improving the quality of video materials has significant potential to increase their value in various fields. These studies require deep knowledge in the areas of signal processing, computer vision, machine learning and understanding the specifics of video data.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1. Існуючі роботи з досліджень з теми видалення шуму	8
2. Теоретичні відомості	10
2.1 Шум у відеоданих	10
2.1.1 Основні типи власних шумів	11
2.1.2 Формули вираховування шумів.....	12
3. Математичні моделі шуму в відеоданих	16
3.1 Типи математичних моделей шуму.....	19
4. Математичні моделі каналів	22
5. Аналіз існуючих методів фільтрації цифрових зображень.....	26
5.1 Загальна характеристика задачі.....	26
5.2 Класифікація методів покращення зображень.....	27
6. Програмна реалізація методів видалення шуму з відеоданих.....	34
Висновки	59
Перелік джерел посилання	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

I – Інтенсивність пікселя в оригінальному зображенні

N – Рівень шуму

I' – Інтенсивність пікселя після обробки

σ^2 – Дисперсія шуму

μ – Середнє значення шуму

CNN – Конволюційна Нейронна Мережа (Convolutional Neural Network)

GAN – Генеративно-Змагальна Мережа (Generative Adversarial Network)

SNR – Відношення Сигнал/Шум (Signal-to-Noise Ratio)

MSE – Середньоквадратична Помилка (Mean Squared Error)

PSNR – Відношення Пікового Сигналу до Шуму (Peak Signal-to-Noise Ratio)

VGG – VGG Мережі (від назви Visual Geometry Group, яка розробила цю архітектуру)

ВСТУП

Процес видалення шуму з відео називається відео-шумопониженням, де шум може бути зменшений на кожному кадрі окремо та між кадрами. Зменшення шуму в послідовності відео широко використовується в управлінні дорожнім рухом, медичному зображенні та телевізійних трансляціях. Зменшення шуму є механізмом відновлення зображення, при якому відновлюється зображення з пошкодженого зображення. Різні техніки шумопониження базуються на різних припущеннях, залежно від типу зображення. У роботі говориться про виявлення руху в потоковому відео в реальному часі за допомогою рухомого фону. Автор розглядав перший кадр як фоновий кадр і порівнював його з поточним кадром, щоб отримати відмінність. Виявлення рухомих об'єктів є першим кроком у відео-шумопониженні, а також відстеженням об'єктів. Ця техніка використовує сегментацію рухомих об'єктів від нерухомих фонових об'єктів. Це спрямовано на обробку більш високого рівня та скорочення часу обчислень. Сегментація тіней об'єктів є складною і важливою через зміни освітлення. Більшість технік відео-шумопониження залежать від одного виду шуму, такого як гаусовий чи білий шум. У роботі було виявлено різні статистичні розподіли з п'ятьма основними джерелами шуму. Це шум від темного струму та квантування.

Коли реальні шумні зображення з різних джерел застосовуються, продуктивність цих технік шумопониження зменшується.

1 ІСНУЮЧІ РОБОТИ З ДОСЛІДЖЕНЬ З ТЕМИ ВИДАЛЕННЯ ШУМУ

На сьогоднішній день існує розширена гама методів для прибирання шумів. Кожен метод призначений для ефективного усування конкретного виду шуму. Існує різні класи фільтрації зображень, які включають в себе різні підходи. Для оптимальної обробки сигналів використовуються різні методи відновлення зображень, такі як лінійні, нелінійні та ітераційні підходи. Розмаїття методів та алгоритмів пов'язане з багатою різноманітністю сценаріїв, які потрібно моделювати різними математичними підходами. Крім того, різні критерії оптимальності використовуються, що призводить до різноманітності методів фільтрації. Навіть коли є спільні моделі та критерії, математичні складнощі часто заважають знайти абсолютно оптимальну процедуру. Ця складність спонукає до використання різних наближених методів і процедур.

У роботі було запропоновано просторовий фільтр у хвильовому домені, коефіцієнти якого обробляються з використанням моделі зображення на основі випадкового поля Маркова (MRF). У літературному огляді автори запропонували просторові (2-вимірні) та просторово-часові (3-вимірні) фільтри, щоб видалити шуми з відео. Просторові фільтри враховують лише просторову інформацію і можуть призводити до просторового розмиття при великому рівні шуму. Цей ефект розмиття можна зменшити, використовуючи як просторову, так і часову інформацію, і таким чином поліпшити фільтрацію при низькому рівні шуму [15].

У хвильовому домені використовується фільтр Вінера для видалення шуму зображення. Автор запропонував фільтр для видалення шуму зображення на основі нечіткої логіки, який враховує відхилення в напрямках. Була запропонована рекурсивна структура для оцінки чистого зображення зі забрудненого шумом від кіноклітин та шум розглядається як пов'язаний з часом експозиції у формі нелінійної та множинної структури. У було

запропоновано піксельний адаптивний просторово-часовий фільтр, який обчислює нові значення пікселів, адаптивно використовуючи зважене середнє значення пікселів на основі компенсованих рухів кадрів. У був представлений фільтр відеошуму, який зберігає контури об'єктів і комбінує фільтри Вінера та Калмана.

Запропонована ідея демонструє нелінійний фільтр відео, який обчислює нові значення пікселів за допомогою 3D вікна. У цьому методі пікселі розташовані відносно відповідних значень пікселів у формі 3D вікна відповідно до їх різниці та усереднюються зважено відповідно до їх порядку сортування. Цей метод дає хороші результати в разі відсутності або повільного локального руху, але може спотворювати області зображення в разі стрімкого локального руху. У разі локального руху 3D фільтрація методу значно знижується. Для підвищення 3D фільтрації методу, запропонованого в, і відеошумопониження використовується 2D та 3D дерев'яний комплексний хвильовий перетворення. Показує фільтрацію на основі 2D хвильового перетворення та фільтрацію на основі середнього часу, яка використовує виявлення руху на основі пікселів [8].

2 ТЕОРЕТИЧНІ ВІДОМОСТІ

Шум виникає в електронних схемах, що формують відеозображення, і викликає похибки при вимірюваннях в СТЗ.

2.1 Шум у відеоданих

Шум у відеоданих – це небажане спотворення зображення, яке може з'являтися під час збору або обробки відео. Існує кілька джерел та процесів шуму:

- електронний шум – це спричинений обмеженнями відеосенсора та електроніки, наприклад, шум від термічних властивостей датчика;
- шум квантування – це процес який виникає під час перетворення аналогових сигналів у цифрові;
- шум стиснення – це процес який утворюється внаслідок втрати даних при стисненні відео;
- шум зовнішнього середовища – це процес який викликаний освітленням, погодними умовами тощо.

Сучасні методи видалення шуму можна розділити на дві великі категорії: класичні алгоритми та методи, засновані на машинному навчанні.

Класичні алгоритми:

- фільтри середнього значення (Mean filtering) згладжують шум шляхом заміни кожного пікселя середнім значенням сусідніх пікселів;
- медіанні фільтри заміняють кожен піксель медіаною сусідніх пікселів, ефективні проти «соль-перець» шуму;
- wiener фільтр використовує статистичний підхід для мінімізації середньоквадратичної помилки між оригінальним та відновленим зображенням.

Методи на основі машинного навчання:

- глибоке навчання (Deep Learning) – це нейронні мережі, особливо конволюційні нейронні мережі (CNN), показали вражаючі результати у видаленні шуму з відеоданих. Моделі, такі як U-Net або VGG, можуть ефективно вчитися на великих наборах даних і відновлювати зображення високої якості;
- передача стилю та генеративно-змагальні мережі (GAN) – саме ці моделі можуть не тільки видаляти шум, але й відновлювати текстури та деталі зображення;
- автоенкодери використовуються для вивчення компактного представлення даних, ефективно відокремлюючи шум від корисного сигналу [23].

Актуальність машинного навчання у цій області полягає у його здатності адаптуватися до різних типів шуму та його ефективності в обробці великих обсягів даних. Завдяки глибокому навчанню, моделі можуть автоматично виявляти складні шаблони в даних та ефективно видаляти шум, не втрачаючи важливих деталей зображення.

2.1.1 Основні типи власних шумів

Основні типи власних шумів в пристроях формування відеозображень: тепловий шум, дробовий шум, шум в фотоелектричних перетворювачах «світло-сигнал».

Розглянемо ймовірнісні моделі випадкових процесів, що відповідають різним типам шумів в електронних схемах.

Тепловий шум – адитивний випадковий процес з нормальним законом розподілу, статистично незалежний від відеосигналу. Спектральна щільність потужності (енергетичний спектр) напруги теплового шуму T – абсолютна температура, R_n – опір навантаження.

Тепловий шум може бути розглянутий як «білий» шум, оскільки його спектральна щільність не змінюється залежно від частоти [5].

Дробовий шум у підсилювальних елементах представляє собою сукупність імпульсів, кожен з яких виникає внаслідок переміщення одного електрона. Струм на виході підсилювального елемента, як сума великої кількості імпульсів, є випадковим процесом і підпорядкованим нормальному розподілу

$$W_{UT} = 4kTR_n,$$

де k – постійна Больцмана, $k = 1,38 \cdot 10^{-23}$ В · с/градус,

T – абсолютна температура,

R_n – опір навантаження.

2.1.2 Формули вираховування шумів

Спектральна щільність потужності струму дробового шуму

$$W_{ID}(\omega) = 2e_{el} I_0 \cdot |S_i(j\omega)|^2,$$

де e_{el} – заряд електрону, $e_{el} = 1,6 \cdot 10^{-19}$ Кл,

I_0 – середнє значення струму на виході підсилювального елемента,

$S_i(j\omega)$ – спектр елементарного імпульсу струму, що виникає внаслідок переміщення заряду одного електрона, нормується.

В смузі частот до $1,6 \cdot 10^{10}$ Гц можна виконати апроксимацію дробового струму «білим» шумом із спектральною щільністю потужності

$$W_{ID} = 2e_{el} I_0.$$

Розглянемо шум у фотоелектричних перетворювачах "світло-сигнал", які застосовуються у сучасних відеокамерах. Фізична природа шуму в фотоемісійних перетворювачах є дробовим у випадку фотопровідникових та фотогальванічних перетворювачів – шум генерації-рекомбінації. При низьких рівнях освітленості фотоелектричного перетворювача, модель шуму можна описати як випадковий процес із розподілом Пуассона. При середніх і високих рівнях освітленості можлива апроксимація шуму нормальним розподілом.

У випадку застосування фотоелектричного перетворювача з великим внутрішнім коефіцієнтом підсилення, такого як фотоелектронний помножувач, дробовий шум переважає над тепловим шумом в підсилювальних схемах. При використанні інших типів перетворювачів, включаючи напівпровідникові перетворювачі «світло-сигнал», тепловий шум переважає в підсилювальних схемах [20].

Для напівпровідникових перетворювачів «світло-сигнал» (ПЗЗ-матриць) можна навести таку класифікацію шумів:

Детерміновані шуми:

- завади від зовнішніх імпульсів керування і живлення;
- геометричні темнові шуми (неоднорідність розподілу темного струму по площі матриці).

Флуктуаційні шуми:

- шум переносу заряду із фоточутливої комірки на вихід ПЗЗ-матриці;
- шум вихідного пристрою ПЗЗ-матриці.

Основний компонент шуму ПЗЗ-матриці – це шум вихідного пристрою (як правило, це підсилювальна схема на польовому транзисторі).

Шум ПЗЗ-матриці визначається за допомогою напруги шуму, яка представляє середньоквадратичне значення флуктуацій часового сигналу на фіксованому опорі навантаження в певному діапазоні частот при відсутності зовнішнього освітлення. Шляхом вимірювання цієї напруги шуму

обчислюється співвідношення сигнал-шум та встановлюється динамічний діапазон для напруги сигналу ПЗЗ-матриці, який становить 50-75 дБ для різних моделей ПЗЗ-матриць.

Звідси можна зробити наступні висновки, враховуючи різні типи шумів у електронних схемах:

- шум у відеосигналі є адитивним випадковим процесом з нормальним розподілом;
- основними складовими шуму, які слід розглядати, є тепловий і дробовий шум електронних схем, включаючи шум вихідного пристрою ПЗЗ-матриці;
- тепловий шум має властивості «білого» шуму і статистично незалежний від корисного сигналу;
- дробовий шум може бути наближено «білим» шумом в діапазоні частот відеосигналу. Він статистично залежить від рівня відеосигналу, проте не має лінійної залежності (збільшення рівня відеосигналу призводить до збільшення дисперсії дробового шуму, але поточне значення напруги шуму може бути будь-яким у визначеному діапазоні);
- для аналітичних розрахунків ймовірнісних характеристик шуму зазвичай використовують непрямі методи, наприклад, значення співвідношення сигнал-шум для відеосигналу на виході пристрою формування відеозображень. Такі розрахунки вимагають складних аналізів електричних схем.

Співвідношення сигнал-шум в телевізійній техніці – це є співвідношення повного динамічного діапазону сигналу яскравості (від рівня чорного $U_{\text{ч}}$ до рівня білого $U_{\text{б}}$) до діючого значення напруги шуму $U_{\text{ш } \delta}$:

$$\Psi = 20 \lg \frac{U_{\text{б}} - U_{\text{ч}}}{U_{\text{ш } \delta}}$$

Оскільки у цій формулі використовується повний динамічний діапазон сигналу яскравості, можна припустити, що яскравість цифрового зображення $Y = Y_{max}$, де Y_{max} – максимально можливе значення яскравості для відеозображення, і шум є статистично незалежним від відеосигналу.

При визначенні ймовірнісних характеристик шуму за допомогою відношення сигнал-шум важливо знайти середньоквадратичне значення його напруги, що є коренем квадратним з дисперсії. У відповідній формулі можна використовувати середньоквадратичне значення напруги шуму замість миттєвого значення, якщо шум є ергодичним випадковим процесом. Ця умова виконується в більшості практичних випадків роботи з відеозображеннями.

Таким чином, вказану формулу можна представити у наступному вигляді:

$$\Psi = 20 \lg \frac{U_b - U_c}{\sigma_{ш}}$$

Для відеосигналу, що виходить з конкретного пристрою (наприклад, відеокамери чи цифрового фотоапарата), відоме відношення сигнал-шум. Таким чином, можна визначити середньоквадратичне значення шуму для цифрового зображення.:

$$\sigma_{ш} = \frac{Y_{max}}{10^{(\Psi/20)}}$$

Для визначення максимального практично можливого значення шуму використовується коефіцієнт амплітуди (пік-фактор). Для шуму з нормальним законом розподілу значення коефіцієнту амплітуди $K_a = 3$ (ймовірність пікових значень шуму, що перевищують максимальне значення, дорівнює 0,3 %) [2].

3 МАТЕМАТИЧНІ МОДЕЛІ ШУМУ В ВІДЕОДАНИХ

Математичні моделі шуму в відеоданих використовуються для опису та симуляції різних типів шуму, які можуть впливати на якість відео. Ці моделі допомагають у покращенні обробки відео, відновленні даних та в розробці алгоритмів зниження шуму. Основні типи шуму в відеоданих включають:

Гауссівський шум – це найпоширеніший тип шуму, який характеризується нормальним розподілом інтенсивності пікселів. Він часто моделюється за допомогою Гауссівської функції, де середнє значення і стандартне відхилення визначають рівень шуму [8].

Шум солі та перцю характеризується різкими, випадковими відмінностями інтенсивності пікселів, зазвичай білими та чорними точками, що нагадують сіль та перець. Шум моделюється як бінарний процес, де кожен піксель незалежно вибирається для внесення шуму з певною ймовірністю.

Пуассонівський шум (або квантовий шум) виникає через варіабельність кількості фотонів, що вдаряють по датчику. Шум має розподіл Пуассона і часто зустрічається в медичному зображенні та астрономічних даних.

Спекл-шум зазвичай зустрічається в радарних і ультразвукових зображеннях і виникає внаслідок взаємодії хвиль. Спекл-шум може бути модельований за допомогою мультиплікативного процесу, що впливає на когерентність хвиль.

Імпульсний шум проявляється у вигляді випадкових сплесків (імпульсів) на відео, які можуть бути спричинені помилками при передачі даних або дефектами датчика.

Математичні моделі цих типів шуму дозволяють розробникам створювати ефективні алгоритми для їх виявлення та видалення, покращуючи якість відеоданих. Наприклад, для зниження шуму можуть використовуватися методи, такі як фільтрація Вінера, медіанна фільтрація, адаптивна фільтрація, та глибоке навчання [10].

Генератор шуму є типовим модулем у синтезаторах, функціонуючим як джерело звуку. Цей генератор створює шум, але на відміну від інших генераторів, які виробляють певні частоти, він випромінює всі частоти хаотично по всьому аудіоспектру. Шум у цьому випадку – це не небажане брязкання, а скоріше равликовий звук, який містить всі частоти у слуховому діапазоні. В синтезаторах часто є вибір між кількома типами шуму, як-от білий та рожевий. Білий шум має рівномірний розподіл усіх частот з однаковою амплітудою в кожній смузі, що призводить до шиплячого або статичного звуку. Рожевий шум має однакову амплітуду в кожній октаві, звучить як помірний вітер.

Шумові сигнали визначаються спектральною плоскістю, де потужність на кожній смузі залежить від частоти у ступенях β . Наприклад, у білому шумі $\beta = 0$. Шумові сигнали класифікують за кольорами, залежно від спектральної щільності, подібно до спектрів різних кольорів видимого світла, що широко використовується в акустиці, електроніці, фізиці та інших областях.

Білий шум має рівномірний спектр на всіх частотах. Він ідеальний для моделювання абсолютно чорного тіла, оскільки включає всі частоти, але його загальна потужність нереально велика. У практичному застосуванні, сигнал вважається «білим», якщо він має рівномірний спектр у певному діапазоні частот.

Термін «білий шум» також відноситься до сигналу з автокореляційною функцією, що математично описується як дельта-функція Дірака. Це означає, що білий шум не корелюється в часі та може мати будь-які значення, за умови що його основна статистична властивість - відсутність кореляції - зберігається. Наприклад, двійковий сигнал, що приймає лише значення 0 або 1, буде білим шумом лише при некорельованій послідовності цих значень. Білий шум може мати різні розподіли, включаючи Гаусса, Пуассона, Коші та інші. Однак гаусівський шум не завжди є білим шумом, оскільки ці поняття відрізняються: гаусівський шум передбачає Гаусове розподіл амплітуд, а «білий» стосується кореляції сигналу в різні моменти часу [15].

Рожевий шум, також відомий як фліккер-шум або $1/f$ -шум, характеризується зниженням спектральної щільності на 3 дБ на кожен октаву зі зростанням частоти, що означає її обернену пропорційність до частоти. Цей шум має рівномірний спектр у логарифмічному масштабі. Один із прикладів рожевого шуму – звук вертольота. Він присутній у таких явищах, як серцеві ритми, мозкова електрична активність та електромагнітне випромінювання космічних об'єктів. Спектральна щільність рожевого шуму описується як приблизно обернено пропорційна частоті, і він є рівномірним у логарифмічній шкалі частот. Наприклад, потужність сигналу між 40 та 60 герцами дорівнює потужності сигналу між 4000 і 6000 герцами, і спектральна щільність знижується на 3 децибелі на кожен октаву порівняно з білим шумом [15].

Червоний шум, іншими словами броунівський шум, має спектральну щільність, пропорційну $1/f^2$. Це означає, що на нижчих частотах його енергія більша, ніж у рожевого шуму, знижуючись на 6 децибел на октаву. Червоний шум має більш приглушений звук порівняно з білим чи рожевим шумами, і його можна отримати інтегруванням білого шуму або за допомогою алгоритму, що імітує броунівський рух [10].

Синій шум, з іншого боку, відрізняється тим, що його спектральна щільність збільшується на 3 дБ на октаву, тобто вона пропорційна частоті. Він сприймається як більш «холодний» або «кислий» звук порівняно з білим шумом і може бути створений шляхом диференціювання рожевого шуму [8].

Фіолетовий шум, який збільшує свою спектральну щільність на 6 дБ на октаву, має спектральну щільність, пропорційну квадрату частоти. Цей шум виникає внаслідок диференціювання білого шуму, а його спектр протилежний до спектру червоного шуму [25].

Сірий шум – це шумовий сигнал, що сприймається людським вухом як однаково гучний у всьому діапазоні частот. Його спектр утворюється шляхом комбінування спектрів броунівського і фіолетового шумів.

Існують також інші, менш відомі види шумів, такі як помаранчевий шум, який має спектр з смугами нульової енергії на частотах музичних нот, або зелений шум, що є схожим на рожевий з акцентом на частоті близько 500 Гц.

3.1 Типи математичних моделей шуму

Розглянемо у деталях типи шуму у відеорядах с прикладами їх вирішення.

Гауссівський шум, який зустрічається у відеоданих, є одним із найпоширеніших типів шуму, що виникає при записі та передачі відео. Цей шум має розподіл інтенсивностей, що слідує нормальному (Гауссівському) закону розподілу, тому інтенсивності пікселів відео відхиляються від їх фактичних значень випадковим чином з деякою середньою та стандартною відхиленням.

Приклади шуму та методи його видалення. Розглянемо на прикладі гауссівської фільтрації.

Уявімо відео з низьким освітленням, де шум стає помітним у вигляді випадкових, візуально неприємних зерен.

Методом видалення буде використання Гауссівського фільтра, який використовує вагові коефіцієнти з Гауссівського розподілу, для згладжування зображення та зменшення видимості шуму.

Розглянемо медіанну фільтрацію, на відеозапису нічного міста, шум може зробити відео «зернистим» і менш чітким.

Методом видалення буде використання медіанного фільтра, де кожен піксель замінюється медіаною значень у його місцевому околі, дозволяє зменшити шум, не розмиваючи гострі краї.

Білатеральний фільтр видаляється наступним чином: у відео з деталізованими текстурями (наприклад, лісовий пейзаж) важливо зберегти деталі при видаленні шуму.

Білатеральний фільтр, який комбінує просторове згладжування з урахуванням різниці в інтенсивності, знижує шум, зберігаючи при цьому гострі краї та текстури буде методом використання.

Вейвлет-фільтрація працює таким чином, як наприклад коли у відео з різкими змінами освітлення, шум може бути більш виражений у певних областях.

За допомогою вейвлет-фільтрації, вона дозволяє розділити відео на різні частотні компоненти та незалежно обробити кожен з них, забезпечуючи більш точне видалення шуму.

Використання машинного навчання стає дедалі частіше. А саме у сучасних відео, які включають складні сцени з високим рівнем деталізації, наприклад, спортивні заходи.

Завдяки алгоритмам машинного навчання та глибокого навчання можуть бути навчені на великих наборах даних для виявлення та видалення шуму, зберігаючи при цьому важливі деталі та текстури.

Ці методи можуть бути застосовані як окремо, так і в комбінації, залежно від типу відеоданих та характеру шуму. Ключ до успішного видалення шуму полягає у виборі правильного балансу між зменшенням шуму та збереженням деталей відео [19].

Пуассонівський шум, також відомий як шум, зумовлений фотонними статистиками, є фундаментальним типом шуму, який часто зустрічається в області обробки зображень, зокрема в медичній візуалізації та астрономічних спостереженнях. Цей шум виникає через варіації в кількості фотонів, які детектуються під час зйомки зображення.

Особливості Пуассонівського шуму проявляються та виникають, коли інтенсивність світла, що падає на датчик, має випадкові варіації, що відповідають Пуассонівському розподілу. Це означає, що в областях з високою інтенсивністю світла спостерігається більше шуму.

Рівень шуму залежить від інтенсивності сигналу. На відміну від Гауссівського шуму, дисперсія Пуассонівського шуму збільшується зі зростанням інтенсивності сигналу.

Шуми можуть проявлятися у різноманітних сферах де використовують відеодані. В астрономічних спостереженнях та медичній візуалізації (наприклад, при використанні ПЕТ або КТ) Пуассонівський шум є значним фактором, що впливає на якість зображень.

На сьогодні існує декілька методів видалення Пуассонівського шуму.

Методи згладжування використовують різні техніки фільтрації для зменшення шуму. Проте, важливо збалансувати процес згладжування так, щоб не втратити важливі деталі зображення.

Використання методів обробки сигналів також розповсюджений метод. Розроблені спеціалізовані алгоритми для видалення Пуассонівського шуму, які враховують його унікальні властивості, наприклад, алгоритми, засновані на вейвлет-перетвореннях або машинному навчанні [8].

Ансамблеві методи в астрономії та інших застосуваннях, де можливе отримання множинних зображень одного й того ж об'єкта, використовують ансамблеві методи, що дозволяють знизити шум шляхом усереднення кількох зображень.

Глибоке навчання застосовує нейронні мережи для видалення шуму є передовою областю досліджень, яка може адаптивно обробляти зображення для видалення Пуассонівського шуму з високою точністю.

Застосування цих методів залежить від конкретних потреб та контексту, в якому виникає Пуассонівський шум. Важливо розуміти, що повне видалення шуму без втрати важливої інформації часто є складним завданням, і потрібен компроміс між зниженням шуму та збереженням деталей [4].

4 МАТЕМАТИЧНІ МОДЕЛІ КАНАЛІВ

В даному розділі розглядаються існуючі математичні моделі каналів зв'язку з білим Гаусовим шумом, аналізуються їх переваги, недоліки та обмеження. Здійснюється аналіз з метою обґрунтування необхідності створення нової математичної моделі, і формується перелік вимог, які, якщо їм відповідає нова модель, дозволять проводити якісні нові дослідження. Завершується розділ розробкою критеріїв для оцінки адекватності математичної моделі.

Проаналізуємо існуючі математичні моделі каналів зв'язку із білим Гауссовим шумом. На сучасний момент, основною математичною моделлю для таких каналів є використання точок сигнального простору. Суть цієї моделі полягає у трансформації вихідного вектора даних (символів вихідного алфавіту модулятора) в точки сигнального простору за вибраним методом модуляції. Після цього на отримані значення накладається білий Гауссів шум, і зашумлені точки сигнального простору перетворюються у вектор вихідних даних (символи вихідного алфавіту модулятора) [20].

Наприклад, для системи багатопозиційної цифрової фазової маніпуляції цей процес можна описати аналітичними формулами, які включають у себе перетворення символу з вихідного алфавіту модулятора в точку сигнального простору:

$$S_i = e^{\frac{2\pi i}{M_j}}$$

де j – уявна одиниця;

M – розмір алфавіту;

i – символ алфавіту модулятора в діапазоні $[0, M - 1]$;

S_i – точка простору сигналу [24].

Ми відзначаємо, що зазначена математична модель ефективно описує процес передачі даних через канали зв'язку з білим Гаусовим шумом і загалом добре підходить для ілюстративних цілей, сприяючи візуалізації переданих та отриманих точок сигнального простору. Однак, навіть при підтвердженій адекватності самої моделі, важливо враховувати, що вона повністю упускає з уваги процеси, які відбуваються в середовищі передачі, а також в передавачі та приймачі, що є суттєвою частиною реальних каналів зв'язку. З одного боку, це сприяє зниженню обчислювальної складності моделювання (іншими словами, програмні реалізації цієї моделі демонструють високу швидкість), але з іншого боку суттєво обмежує область її застосування [12].

Як альтернативу, розглядається інтуїтивна векторна математична модель каналу зв'язку із білим Гаусовим шумом, що базується на уявленні тимчасових процесів у каналах зв'язку у формі тимчасових векторів, а саме серій відліків з тимчасовими мітками. Незважаючи на те, що ця математична модель виглядає досить простою, на даний момент відсутній повноцінний формальний опис, і існуючі реалізації є специфічними та часто придатними лише для конкретних типів каналів зв'язку.

В результаті аналізу існуючих математичних моделей каналів зв'язку із білим Гаусовим шумом виникла потреба у новій моделі, яка дозволяє досліджувати та моделювати не лише загальний процес передачі даних, але й фізичні процеси, що відбуваються в окремих компонентах каналу (передавач, середовище передачі, приймач).

Враховуючи це, формулюються наступні вимоги до розроблюваної математичної моделі:

- адекватність – математична модель повинна точно відображати реальність, а отримані з неї дані повинні відповідати як теоретичним, так і емпіричним даним інших адекватних математичних моделей;
- реалізованість – повинна бути ефективно реалізована за допомогою сучасних технічних засобів, наприклад, комп'ютерних програм, і надавати результат за прийнятний час;

- застосовність для моделювання процесу передачі даних в цілому повинна бути застосовною для вивчення можливостей бітових і/або символічних помилок для різних способів модуляції даних;
- моделювання процесів в передавачі – ця модель повинна включати передавач, що перетворює вхідні дані в сигнали для передачі та дозволяє спостерігати та впливати на процеси в ньому;
- моделювання процесів в середовищі передачі – включає середовище передачі даних, дозволяючи спостерігати та впливати на процеси, що відбуваються в ньому;
- моделювання процесів в приймачі – ця модель може включати приймач, який розпізнає та декодує зашумлені сигнали, дозволяючи спостерігати та впливати на процеси в ньому;
- розширюваність – модель повинна мати гнучку структуру, що дозволяє розширювати її для застосування в різних завданнях.

Враховуючи останню вимогу, що математична модель, яку ми розробляємо, має бути придатною для широкого спектру завдань, слід зауважити, що згідно з 4-м і 5-м пунктами вищезазначених вимог стає очевидним, що така математична модель не може бути абсолютно універсальною. Іншими словами, не існує аналітичного виразу, за допомогою якого можна було б повністю описати бажану математичну модель так, щоб вона була застосовною для будь-якого завдання без будь-яких додаткових налаштувань чи адаптацій.

У зв'язку з цим, можливі варіанти реалізації можуть бути обмежені певними типами приймачів і передавачів. Проте ці реалізації повинні включати можливості для розширення шляхом включення аналітичних виразів для інших типів приймачів і передавачів, з урахуванням збереження відповідності розмірностей і порядків величин [19].

Щоб визначити адекватність математичної моделі каналу зв'язку з білим Гаусовим шумом, встановлюються наступні критерії:

- візуалізація процесів необхідна подання процесів в передавачі, лінії зв'язку та приймачі повинні відповідати очікуваним теоретичним уявленням про ці процеси;
- ймовірності символної помилки побудовані за допомогою математичної моделі, повинні відповідати очікуваним теоретичним даним.

5 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ФІЛЬТРАЦІЇ ЦИФРОВИХ ЗОБРАЖЕНЬ

5.1 Загальна характеристика задачі

У завданнях пов'язаних з передачею, обробкою та зберіганням зображень, велике значення мають методи стиснення даних. Це особливо актуально через те, що дані в таких системах, зазвичай, представляють собою інтенсивні цифрові потоки. Тому навіть невелике скорочення обсягу інформації допомагає зменшити вимоги до пропускної здатності та обсягу зберігання даних. У цьому контексті, якість роботи алгоритмів стиснення має прямий вплив на ефективність функціонування системи. При цьому важливу роль відіграють зображення та відеопослідовності, які становлять значну частину оброблюваної інформації [13].

Ефективним методом зменшення інтенсивності цифрового потоку є стиснення передачі даних з можливістю подальшого відновлення на стороні отримувача. Традиційні методи стиснення без втрат зазвичай відзначаються обмеженою ефективністю при роботі з зображеннями. Тому часто використовуються методи стиснення з втратами, які дозволяють досягти більш високого ступеня стиснення. Однак, важливо, щоб стиснуті зображення зберігали високу візуальну якість [2]. Наприклад, для успішного вирішення завдань пошуку та ідентифікації об'єктів та визначення їх характеристик, важливо, щоб зображення залишалися високою візуальною якістю, навіть після стиснення з втратами, а також у відмінних умовах отримання зображень, при використанні недосконалих систем передачі відеоінформації та її відтворенні, а також в умовах впливу різних спотворень і завад.

Отже, проблема видалення шуму є однією з найактуальніших і поширених проблем в області оброблення як статичних зображень, так і відео.

5.2 Класифікація методів покращення зображень

Основною ціллю вдосконалення є така обробка зображення, яка забезпечить найкращий результат для певного використання. Наприклад, техніка, що ефективно покращує рентгенівські знімки, може не бути ідеальним рішенням для обробки фотографій, зроблених космічним апаратом. Існує велика різноманітність методів вдосконалення зображень, які можна класифікувати на дві головні категорії: методи, які використовуються у просторовій області, та методи, які застосовуються у частотній області. Термін «просторова область» стосується площини зображення і включає підходи, які безпосередньо впливають на пікселі. З іншого боку, методи, що працюють у частотній області, базуються на зміні сигналів через застосування перетворення Фур'є. Також існують технології, що інтегрують різні методи з цих двох категорій. Просторові методи включають процедури, які маніпулюють безпосередніми значеннями пікселів, а процеси просторової обробки можна описати певними рівняннями.

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

де $f(x, y)$ – вхідне зображення;

$g(x, y)$ – оброблене зображення T – оператор над f , визначений в деякому околі точки (x, y) .

Основний підхід по визначенню меж точки (x, y) , полягає у використанні квадратної або прямокутної області – підмножини зображень, центрованої в точці (x, y) . Її центр пересувається від пікселя до пікселя, починаючи з верхнього лівого кута. Оператор T виконується в кожній точці (x, y) , даючи в результаті вихідне значення g для даної точки. Процес використовує лише пікселі в середині області зображення, обмеженою певною ділянкою. Квадратні або прямокутні масиви є, безумовно, найбільш розповсюдженими

завдяки простоті їх реалізації, хоча іноді використовуються ділянки інших форм, наприклад ті, що наближаються до кола.

Найпростіша форма оператора T досягається у випадку, коли область має розміри 1 на 1 (тобто один піксель). В такому випадку g залежить лише від значення f в точці (x,y) , і T стає функцією градаційного перетворення, що також має назву «функції перетворення інтенсивності» або «функції відображення» вигляду:

$$s = T(r) f(x,y) + g(x,y)$$

де r і s – змінні, що визначають відповідні значення яскравостей зображень $f(x,y)$ та $g(x,y)$ в кожній точці (x,y) [4].

Так як якість покращення кожного окремого елемента зображення залежить від його світлості, методи цієї категорії зазвичай класифікуються як процедури обробки окремих елементів. Розширення радіуса області призводить до більшої гнучкості. Ідея полягає у використанні значень функції f у певному радіусі навколо точки (x,y) для обчислення нового значення g в цій точці. Одним з ключових методів є використання так званих масок (або фільтрів, ядер, шаблонів чи вікон), які зазвичай є невеликими (часто 3×3 елементів) двовимірними масивами. Коефіцієнти цих масок визначають характер обробки, наприклад, підвищення чіткості зображення. Методи, що використовують цей підхід, часто називають обробкою маскою або фільтрацією за маскою [4].

Процедура фільтрації в частотній області досить проста та включає такі кроки (рис 5.1):

- початкове зображення перемножується на $(-1)^{x+y}$, щоб забезпечити центрування його Фур'є-перетворення;
- обчислюється пряме перетворення Фур'є $F(u,v)$ для отриманого зображення після першого кроку;

- функція $F(u, v)$ множиться на функцію фільтра $H(u, v)$;
- обчислюється обернене перетворення Фур'є для отриманого результату з кроку 3;
- виділяється дійсна частина результату з кроку 4;
- результат кроку 5 множиться на $(-1)^{x+y}$.

Численні приклади свідчать про те, що деякі завдання з удосконалення зображень, вирішення яких в просторовій області є вкрай складним або навіть неможливим, стають майже тривіальними, якщо використовувати частотну область. Після вибору характерного фільтра експериментально в частотній області, зазвичай можлива реалізація відповідного методу в просторовій області.

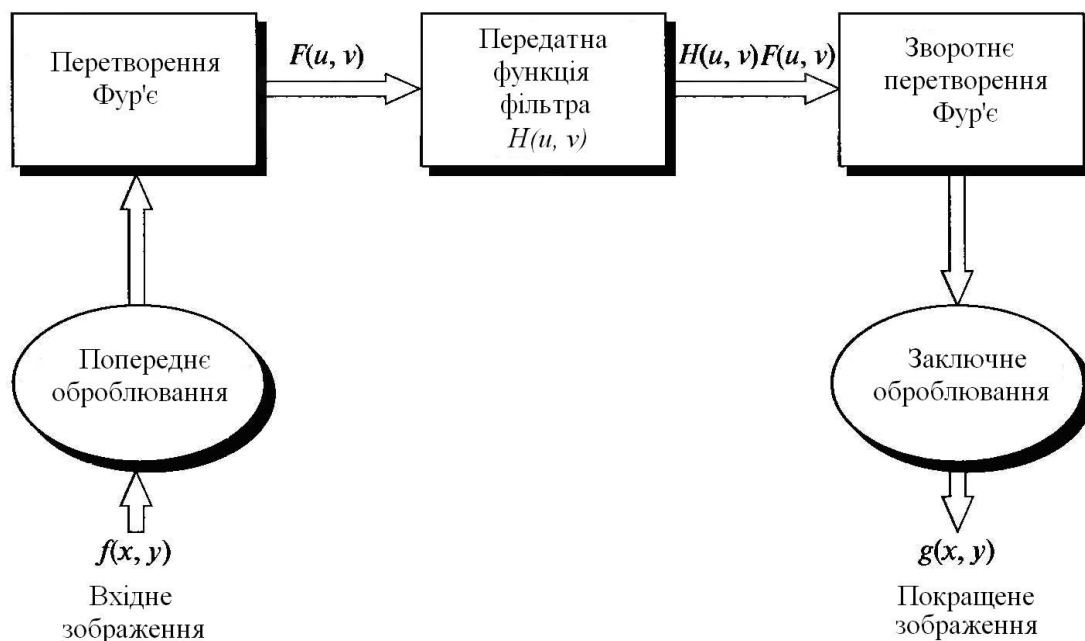


Рисунок 5.1 – Основні етапи фільтрації в частотній області

Одним з ключових зв'язків між просторовою та частотною областями у фільтрації є відомий результат, відомий як "теорема про згортку". Ця теорема базується на операції згортки, де певна маска переміщується по зображенню від елемента до елемента, і для кожного елемента обчислюється заздалегідь визначена величина.

В наш час недостатньо лише згладжувати значення u для отримання чистого зображення. Сучасні методи не лише згладжують зображення, але також передбачають відновлення втраченої інформації $n(D_h, u)$, якщо це необхідно. Це означає, що при отриманні зображення за допомогою цифрової дзеркальної камери часто важливо зберігати чіткість та деталі, тоді як шум необхідно ефективно розмити.

Локальні методи фільтрації можуть бути використані з застосуванням спеціальних фільтрів. Оригінальне (півтонове) зображення u визначається в обмеженій області $\Omega \subset R^2$ та позначається як $u(x)$ для $x \in R^2$. Відповідно до звичайної практики візуалізації на екрані або на принтері інтерполуються дискретні значення шуму n_i не як функція з обмеженим спектром, а як кусочно-постійна функція, постійна в кожному пікселі i та рівна n_i . Можна записати x як L^2 норму та $x \cdot y$ як скалярний добуток.

Фільтр Гауса – найбільш використовуваний фільтр розмиття, який дійсноє згортку зображення лінійним симетричним ядром. Необхідність гладжування зазвичай виражається за допомогою позитивності ядра.

Формула такого ядра також називається ядром Гауса

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

Функція має середньоквадратичне відхилення. Обчислення правильне якщо h достатньо мале. З іншого боку, властивості зменшення шуму залежать від того факту, що околиці, які приймають участь у згладжуванні достатньо великі, таким чином шум зменшується шляхом осереднення. В подальшому, якщо припустити, що $h = k\epsilon$, де k – кількість відрахунків функції u та шуму в інтервалі довжиною h , ϵ^2 локального вікна, k повинно бути більше за 1 [3].

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

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

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

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

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

Фільтри околів приймають до уваги значення рівнів сірого для визначення сусідніх пікселів. В такому випадку очищене від шуму значення пікселя i є зваженим середнім значень у пікселях, які мають значення рівня сірого близьке до $u(i)$. Можна визначити окіл рівня сірого, як

$$U(i, h) = \{j \in I \mid u(i) - h < u(j) < u(i) + h\}$$

Таким чином, це також локальна схема в області інтенсивності (яскравості). Але вона нелокальна в просторовій області, оскільки пікселі, які належать всьому зображенню використовуються для обчислення в

пікселі i . Популярним варіантом цього алгоритму є середньозважений фільтр

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

де $\Omega \subset R^2$ – відкрита та обмежена множина;

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

Розглянемо формулу коефіцієнту нормалізації. Фільтри околів розглядають змішані околи $U(i, h) \cap B_{\rho}(i)$, деколо з центром i та радіусом ρ [10].

Білатеральний фільтр (БФ) вперше представлений у 1998 році, використовує метод просторового зваженого усереднення, зберігаючи при цьому чіткість границь. Це досягається за допомогою двох типів гаусівських фільтрів: один фокусується на просторовому аспекті, тоді як інший – на інтенсивності або яскравості. Важливість як просторової відстані, так і відстані інтенсивності визначається в процесі фільтрації [18].

Білатеральний фільтр – це нелінійний, краєзберігаючий та шумопоглинаючий метод фільтрації зображень. Він ефективний для видалення шуму, зберігаючи при цьому гостроту країв. Це робить білатеральний фільтр особливо корисним у застосуваннях, де важливі як шумоподавлення, так і збереження деталей зображення [6].

Хоча білатеральний фільтр широко використовується, відсутнє конкретне теоретичне обґрунтування для вибору розміру його вікна. Цей

параметр зазвичай визначається експериментально. Відомо, що малий розмір вікна обмежує ефективність зниження шуму через використання обмеженої кількості взаємопов'язаних пікселів. Навпаки, великий розмір вікна, що включає більше непов'язаних пікселів, може негативно вплинути на якість знешумлення, роблячи алгоритм менш ефективним.

Білатеральний фільтр враховує не тільки просторову (тобто, геометричну чи просторову близькість), але й яскравісну (інтенсивність або колір) близькість.

Це означає, що для кожного пікселя фільтр вагає сусідні пікселі не тільки на основі їх просторової близькості до розглядуваного пікселя, але й на основі подібності їх яскравості або кольору.

Вагові коефіцієнти:

- фільтр використовує два різні види вагових коефіцієнтів: один для просторової близькості (зазвичай гауссівський) та інший для яскравісної близькості;

- просторовий ваговий коефіцієнт зменшується зі збільшенням відстані від центрального пікселя;

- яскравісний ваговий коефіцієнт зменшується з різницею інтенсивності. Це дозволяє зберегти краї, оскільки великі різниці в інтенсивності (як на краях) приведуть до низьких ваг.

Завдяки цій двокомпонентній ваговій системі, білатеральний фільтр може ефективно зменшувати шуми, не розмиваючи краї [17].

6 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ВИДАЛЕННЯ ШУМУ З ВІДЕОДАНИХ

В сучасному світі обробка відеоданих виявляється надзвичайно актуальною та важливою задачею в багатьох галузях, включаючи медицину, безпеку, розваги та багато інших. Однак, разом із зростанням обсягів відеоінформації, виникає проблема ефективної обробки та аналізу великих обсягів даних. Шум у відеоданих є однією з ключових проблем, яка може суттєво впливати на якість та точність подальшого аналізу даних.

Розділ, присвячений програмній реалізації методів видалення шуму з відеоданих, є важливою складовою дипломної роботи. Цей розділ спрямований на вирішення завдань підвищення якості та достовірності відеоінформації, забезпечуючи ефективне видалення шумових елементів, що виникають внаслідок різних факторів, таких як артефакти компресії, непередбачені умови зйомки, або джерела зовнішнього шуму.

Цей розділ пропонує висвітлити ключові аспекти програмної реалізації методів видалення шуму, зосереджуючись на виборі та оптимізації алгоритмів, розробці програмних інструментів та їхній інтеграції в відповідні системи обробки відеоданих. Також буде розглянуто сучасні підходи до реалізації, враховуючи останні досягнення в галузі машинного навчання та комп'ютерного зору.

Основні завдання розділу включають аналіз і вибір оптимальних методів видалення шуму для конкретного контексту, розробку програмної архітектури та інтерфейсу для взаємодії з алгоритмами, а також експериментальне порівняння та оцінку ефективності розроблених методів на реальних відеоданих. Додатково, розділ розгляне питання оптимізації та прискорення алгоритмів видалення шуму для використання в реальному часі.

Загальна мета цього розділу – представити наглядний та базовий приклад розуміння методів видалення шуму у відеоданих та їхню ефективну програмну реалізацію, враховуючи високі вимоги до якості та продуктивності обробки великих обсягів відеоданих.

Розглянемо декілька популярних методів видалення шуму з відеоданих.

Фільтр Гауса

Для створення програми, що видаляє шум з відеоданих за допомогою гауссового фільтра в Python, ви можете використовувати бібліотеки OpenCV та NumPy. Ось основні кроки, які потрібно виконати у програмі:

- імпортувати необхідні бібліотеки;
- зчитати відеофайл;
- перебирати кожен кадр відео;
- застосувати гауссівський фільтр до кожного кадру для згладжування та видалення шуму;
- записати оброблений кадр у вихідний відеофайл;
- закрити всі відкриті вікна та вихідні файли;

Цей код відкриває відео, застосовує гауссівський фільтр до кожного кадру, а потім записує результат у новий відеофайл. Ви можете змінити розмір ядра фільтра та стандартне відхилення (у цьому прикладі (5, 5) і 0 відповідно) для досягнення бажаного ефекту.

Лістинг 6.1 Реалізація застосування гауссівського фільтра:

```
import cv2

import numpy as np

def remove_noise(input_video_path, output_video_path):
    # Зчитування відео
    cap = cv2.VideoCapture(input_video_path)
```

```
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter(output_video_path, fourcc, 20.0, (int(cap.get(3)),
int(cap.get(4))))

while (cap.isOpened()):
    ret, frame = cap.read()
    if ret:
        # Застосування гауссового фільтра
        filtered_frame = cv2.GaussianBlur(frame, (5, 5), 0)

        # Запис обробленого кадру
        out.write(filtered_frame)

        # Відображення кадру
        cv2.imshow('frame', filtered_frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

# Закриття всіх вікон
cap.release()
out.release()
cv2.destroyAllWindows()

# Вказівка шляху до вхідного та вихідного відеофайлів
input_video = r'C:\Users\enCore\Desktop\Video\your_video.mp4'
```

```
output_video = r'C:\Users\enCore\Desktop\denoised_video.avi'  
remove_noise(input_video, output_video)
```

Видалення шуму з відео за допомогою гауссового фільтра включає декілька ключових кроків та принципів. Ось докладний опис цього процесу:

Читання Відео та зчитування кадрів. Програма спочатку відкриває відеофайл та зчитує його кадр за кадром. Це робиться за допомогою функцій бібліотеки OpenCV.

Застосування Гауссового Фільтра – це тип згладжуючого фільтра, який використовує ядро фільтра з гауссівським розподілом (крива дзвона) для видалення шуму.

Ядро фільтра – це матриця, яка «накладається» на пікселі зображення. Для кожного пікселя розраховується взважене середнє значення пікселів, що його оточують, з вагами, визначеними гауссовим ядром.

Розмір ядра (зазвичай нечетне число, наприклад 5x5 або 3x3) впливає на ступінь згладжування. Більші ядра виробляють сильніше згладжування, але можуть зробити зображення занадто «розмитим».

У гауссовому фільтрі можна вказати стандартне відхилення в гауссівській формулі. Воно контролює, наскільки широкою буде крива дзвона.

Після обробки кожного кадру згладженим фільтром, кадр записується у новий відеофайл.

Відображення кадрів виконується опційно, оброблені кадри можна відображати в реальному часі під час обробки.

Після обробки всіх кадрів відео зберігається у вказаному місці.

Треба зробити важливі зауваження, що гауссовий фільтр ефективний для видалення гауссівського шуму, який має нормальний розподіл інтенсивності. Фільтр може не бути таким ефективним для інших типів шуму, наприклад, сольово-перцевого шуму.

Сильне згладжування може призвести до втрати важливих деталей у відео.

Цей метод є одним із базових підходів до видалення шуму та може бути доповнений або замінений більш складними методами, залежно від конкретних вимог до якості та типу відеоданих.

Прикладом буде фрагмент відео де літак здійснює посадку:



Рисунок 6.1 – Порівняльний результат видалення Гауссівського шуму

На зображенні ви можете побачити, як виглядає фотографія літака до та після видалення гауссівського шуму. Ліва частина зображення показує літак з гауссівським шумом, що виглядає як випадкова, зерниста текстура на зображенні, представляючи стан «до» обробки. Права частина зображує той самий літак після видалення гауссівського шуму за допомогою гауссівського фільтра, що призводить до чистішого та гладшого вигляду, представляючи стан «після» обробки. Перехід між цими двома станами чітко демонструє ефект видалення шуму.

Видалення Пуассонівського шуму з відеоданих є трохи складнішою задачею, оскільки Пуассонівський шум (або фотонний шум) залежить від інтенсивності пікселя і зазвичай виникає в умовах низької освітленості. Для видалення Пуассонівського шуму часто використовують методи, що базуються на моделюванні шуму, як-то варіаційні методи, але також ефективними можуть бути деякі простіші підходи, такі як медіанний фільтр або фільтр Гаусса, залежно від контексту.

Один з найпоширеніших методів – це використання фільтра Гаусса, який може допомогти зменшити Пуассонівський шум, особливо якщо шум не дуже високий. Ось приклад програми на Python, яка використовує фільтр Гаусса для видалення шуму з відео:

Лістинг 6.2 Приклад програми на Python для використання фільтра:

```
import cv2

def remove_poisson_noise(input_video_path, output_video_path):
    cap = cv2.VideoCapture(input_video_path)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter(output_video_path, fourcc, 20.0, (int(cap.get(3)),
int(cap.get(4))))

    while cap.isOpened():
        ret, frame = cap.read()
        if ret:
            # Застосування фільтра Гаусса
            filtered_frame = cv2.GaussianBlur(frame, (5, 5), 0)

            out.write(filtered_frame)
```

```

cv2.imshow('frame', filtered_frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
else:
    break

cap.release()
out.release()
cv2.destroyAllWindows()

# Використання функції
input_video = r'C:\Users\enCore\Desktop\Video\your_video.mp4'
output_video = r'C:\Users\enCore\Desktop\denoised_video.avi'
remove_poisson_noise(input_video, output_video)

```

Наведемо деякі приклади використання Фільтра Гаусса:

- фільтр Гаусса ефективний для згладжування зображень, він може допомогти зменшити Пуассонівський шум, але також може зробити деякі деталі зображення менш різкими;

- параметри фільтра (розмір ядра та стандартне відхилення) можуть бути налаштовані для досягнення оптимального балансу між зниженням шуму та збереженням деталей;

Давайте тепер розглянемо зображення, щоб продемонструвати, як ця програма може вплинути на зображення із Пуассонівським шумом.

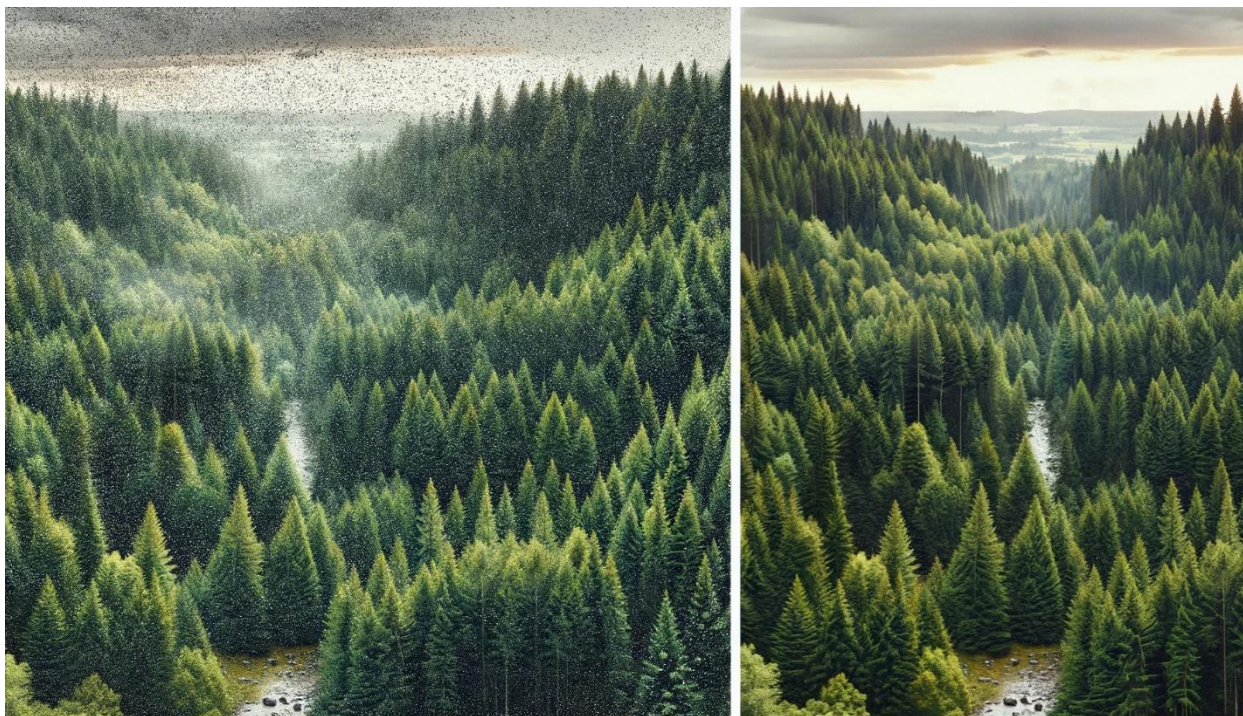


Рисунок 6.2 – Порівняльний результат видалення Пуассонівського шуму

На зображенні ви можете побачити приклад використання методів зниження шуму для видалення Пуассонівського шуму з фотографії лісового пейзажу. Ліва сторона зображення показує лісовий пейзаж з видимим Пуассонівським шумом, що має вигляд випадкової, плямистої текстури, представляючи стан «до» обробки. Права сторона зображення показує той самий лісовий пейзаж після застосування методів зниження шуму, де Пуассонівський шум значно зменшений, що призводить до чистішого та детальнішого вигляду, представляючи стан «після» обробки.

Щоб створити програму на Python для видалення шуму з відеоданих за допомогою білатерального фільтра, вам знадобиться використовувати бібліотеку OpenCV. Нижче наведений приклад такої програми:

Лістинг 6.3 Приклад коду для застосування білатерального фільтра:

```
import cv2
import numpy as np

def remove_noise_with_bilateral_filter(input_video_path, output_video_path):
    # Зчитування відео
    cap = cv2.VideoCapture(input_video_path)

    # Визначення кодеку та створення об'єкта VideoWriter
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter(output_video_path, fourcc, 20.0, (int(cap.get(3)),
int(cap.get(4))))

    while cap.isOpened():
        ret, frame = cap.read()
        if ret:
            # Застосування білатерального фільтра
            filtered_frame = cv2.bilateralFilter(frame, 9, 75, 75)
            # Запис обробленого кадру
            out.write(filtered_frame)
            # Відображення кадру
            cv2.imshow('frame', filtered_frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
        else:
            break

    # Закриття всіх вікон
```

```
cap.release()  
out.release()  
cv2.destroyAllWindows()
```

Використання функції

```
input_video = r'C:\Users\enCore\Desktop\Video\your_video.mp4'  
output_video = r'C:\Users\enCore\Desktop\denoised_video.avi'  
remove_noise_with_bilateral_filter(input_video, output_video)
```

Перед запуском програми, переконайтеся, що у вас встановлена бібліотека OpenCV. Якщо вона ще не встановлена, ви можете встановити її за допомогою `pip`:

```
pip install opencv-python
```

Для тесту цієї програми ми візьмемо кадр з відео з горами.



Рисунок 6.3 – Порівняльний результат застосування білатерального фільтра

Це зображення, що демонструє ефект використання білатерального фільтра для видалення шуму з фотографії природного ландшафту. На лівій стороні зображення ви можете побачити ландшафт з видимим шумом, який виглядає як зерниста текстура, що представляє стан «до» обробки. На правій стороні зображено той самий ландшафт після застосування білатерального фільтра, де шум значно зменшено, що призводить до чистішого та детальнішого вигляду, представляючи стан «після» обробки.

Програма для видалення шуму з відеоданих, працює наступним чином:

Перш за все, програма імпортує необхідні бібліотеки: OpenCV (cv2) для обробки відео та зображень, та NumPy для операцій з матрицями, хоча в даному випадку NumPy може й не використовуватися безпосередньо.

Функція `remove_noise_with_bilateral_filter` – ця функція виконує основну роботу по видаленню шуму з відео:

- функція відкриває відеофайл для читання з використанням `cv2.VideoCapture`;
- шлях до вхідного відеофайлу передається як аргумент `input_video_path`;
- створюється об'єкт `VideoWriter` для запису вихідного відео;
- використовується той же кодек та розміри, що й у вхідному відео (за допомогою `cap.get`);

Етап обробки кадрів проходить таким чином:

- виконується цикл для читання кадрів відео один за одним;
- для кожного кадра застосовується білатеральний фільтр з використанням `cv2.bilateralFilter`. Параметри фільтра (`diameter`, `sigmaColor`, `sigmaSpace`) визначають ступінь згладжування та збереження країв.

Наступний етап – це запис та відображення кадрів

- кожен оброблений кадр записується у вихідний відеофайл;
- опційно, кадри можуть відображатися під час їх обробки.

Після обробки всіх кадрів відео, ресурси, що використовуються (відео вхід/вихід), звільняються, та закриваються всі відкриті вікна OpenCV.

Фінальним етапом буде виклик функції `remove_noise_with_bilateral_filter` викликається зі шляхами до вхідного та вихідного файлів. Вхідний файл береться з `C:\Users\enCore\Desktop\Video\your_video.mp4`, а результат зберігається на `C:\Users\enCore\Desktop\denoised_video.avi`.

Необхідно підкреслити наступні зауваження:

- під час роботи програми, важливо підібрати оптимальні параметри для білатерального фільтра, оскільки вони впливають як на ефективність шумоподавлення, так і на збереження деталей зображення;
- білатеральний фільтр може бути обчислювально вимогливим, особливо для великих відеофайлів або високих роздільних здатностей, тому слід враховувати час обробки.

Фільтри околів (медіанні фільтри) є ще одним популярним методом для видалення шуму з відеоданих, зокрема ефективним проти так званого «сольово-перцевого» шуму. Нижче наведено опис та приклад програми на Python, яка використовує медіанний фільтр для видалення шуму з відео.

Як працює Медіанний Фільтр

Основний принцип роботи – медіанний фільтр замінює кожен піксель зображення медіаною інтенсивностей пікселів у своєму сусідстві (околі).

Около Пікселя або «около» – це зазвичай квадрат (наприклад, 3x3, 5x5 пікселів), який переміщується по зображенню. Для кожного пікселя в зображенні розглядається набір пікселів, що входять до цього околу.

Наступним елементом буде медіана – це середній елемент впорядкованого за значенням набору чисел. У випадку медіанного фільтра, візьмуться інтенсивності всіх пікселів у околі, вони впорядковуються, і медіанне значення встановлюється як нове значення центрального пікселя.

Збереження країв на відміну від лінійних фільтрів, медіанний фільтр добре зберігає краї об'єктів на зображенні.

Лістинг 6.4 Реалізація в Python з використанням OpenCV:

```
import cv2

def remove_noise_with_median_filter(input_video_path, output_video_path):
    cap = cv2.VideoCapture(input_video_path)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter(output_video_path, fourcc, 20.0, (int(cap.get(3)),
int(cap.get(4))))

    while cap.isOpened():
        ret, frame = cap.read()
        if ret:
            # Застосування медіанного фільтра
            filtered_frame = cv2.medianBlur(frame, 5) # Використовуючи около
5x5

            out.write(filtered_frame)

            cv2.imshow('frame', filtered_frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
        else:
            break

    cap.release()
    out.release()
    cv2.destroyAllWindows()
```

Використання функції

```
input_video = r'C:\Users\enCore\Desktop\Video\your_video.mp4'
```

```
output_video = r'C:\Users\enCore\Desktop\denoised_video.avi'
```

```
remove_noise_with_median_filter(input_video, output_video)
```

Використання медіанного фільтра є особливо ефективним для видалення «сольово-перцевого» шуму.

Розмір околу (наприклад, 3x3, 5x5) може бути налаштований залежно від ступеня шуму в зображенні.



Рисунок 6.4 – Порівняльний результат застосування медіанного фільтра

На зображенні ви можете побачити приклад використання медіанного фільтра для видалення шуму з фотографії міського пейзажу. Ліва сторона зображення показує міський пейзаж із видимим шумом, що має вигляд випадкової, зернистої текстури, представляючи стан «до» обробки. Права

сторона зображення показує той самий міський пейзаж після застосування медіанного фільтра, де шум значно зменшений, що призводить до чистішого та детальнішого вигляду, представляючи стан «після» обробки.

Видалення Пуассонівського шуму з відеоданих є трохи складнішою задачею, оскільки Пуассонівський шум (або фотонний шум) залежить від інтенсивності пікселя і зазвичай виникає в умовах низької освітленості. Для видалення Пуассонівського шуму часто використовують методи, що базуються на моделюванні шуму, як-то варіаційні методи, але також ефективними можуть бути деякі простіші підходи, такі як медіанний фільтр або фільтр Гаусса, залежно від контексту.

Один з найпоширеніших методів – це використання фільтра Гаусса, який може допомогти зменшити Пуассонівський шум, особливо якщо шум не дуже високий. Ось приклад програми на Python, яка використовує фільтр Гаусса для видалення шуму з відео.

Лістинг 6.5 Реалізація видалення пуассонівського шуму:

```
import cv2

def remove_poisson_noise(input_video_path, output_video_path):
    cap = cv2.VideoCapture(input_video_path)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter(output_video_path, fourcc, 20.0, (int(cap.get(3)),
int(cap.get(4))))

    while cap.isOpened():
        ret, frame = cap.read()
        if ret:
            # Застосування фільтра Гаусса
            filtered_frame = cv2.GaussianBlur(frame, (5, 5), 0)
```

```

    out.write(filtered_frame)

    cv2.imshow('frame', filtered_frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    else:
        break

cap.release()
out.release()
cv2.destroyAllWindows()

# Використання функції
input_video = r'C:\Users\enCore\Desktop\Video\your_video.mp4'
output_video = r'C:\Users\enCore\Desktop\denoised_video.avi'
remove_poisson_noise(input_video, output_video)

```

Наступним методом видалення шуму розглянемо вейвлет-фільтрацію для видалення шуму. Вейвлет-фільтрація є потужним методом для видалення шуму з зображень та відео. Вона використовує вейвлет-перетворення для розділення зображення на різні частотні компоненти, а потім модифікує ці компоненти для зменшення шуму.

Процес Вейвлет-фільтрації полягає у таких аспектах:

- вейвлет-Перетворення – зображення перетворюється з просторового домену в вейвлет-домен, розділяючи деталі зображення на різні рівні (частоти);

– обробка коефіцієнтів вейвлетів, які представляють високочастотні деталі (і, зазвичай, шум), модифікуються або відсікаються. Це може включати порогову обробку коефіцієнтів;

– зворотне вейвлет-перетворення обробки коефіцієнтів виконується, щоб перевести зображення назад у просторовий домен.

Для реалізації вейвлет-фільтрації у Python, можна використовувати бібліотеку PyWavelets. Ось приклад програми, яка застосовує вейвлет-фільтрацію до кадрів відео.

Лістинг 6.6 Реалізація вейвлет-фільтрації з використанням Python:

```
import cv2
import pywt
import numpy as np

def wavelet_denoising(frame):
    # Розділення на канали кольорів
    coeffs2 = pywt.dwt2(frame, 'haar')
    LL, (LH, HL, HH) = coeffs2
    # Модифікація коефіцієнтів
    LH = np.zeros_like(LH)
    HL = np.zeros_like(HL)
    HH = np.zeros_like(HH)
    # Зворотне перетворення
    coeffs2 = LL, (LH, HL, HH)
    return pywt.idwt2(coeffs2, 'haar')

def remove_noise_with_wavelet_filter(input_video_path, output_video_path):
    cap = cv2.VideoCapture(input_video_path)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
```

```

out = cv2.VideoWriter(output_video_path, fourcc, 20.0, (int(cap.get(3)),
int(cap.get(4))))

```

```

while cap.isOpened():
    ret, frame = cap.read()
    if ret:
        # Застосування вейвлет-фільтрації до кадру
        filtered_frame = wavelet_denoising(frame)

        out.write(filtered_frame)

        cv2.imshow('frame', filtered_frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

cap.release()
out.release()
cv2.destroyAllWindows()

```

```

# Використання функції
input_video = r'C:\Users\enCore\Desktop\Video\your_video.mp4'
output_video = r'C:\Users\enCore\Desktop\denoised_video.avi'
remove_noise_with_wavelet_filter(input_video, output_video)

```

Розглянемо приклад зображення щоб продемонструвати ефект вейвлет-фільтрації на фотографії використовуючи цю програму.



Рисунок 6.5 – Використання ефекту вейвлет-фільтрації

На зображенні ви можете побачити ефект використання вейвлет-фільтрації для видалення шуму з фотографії морського пейзажу. Ліва сторона зображення показує морський пейзаж з видимим шумом, що має вигляд випадкової, зернистої текстури, представляючи стан «до» обробки. Права сторона зображення показує той самий морський пейзаж після застосування вейвлет-фільтра, де шум значно зменшений, що призводить до чистішого та детальнішого вигляду, представляючи стан «після» обробки.

Програма для видалення шуму з відео за допомогою вейвлет-фільтрації працює на основі декількох ключових кроків. Ось детальний опис її роботи:

Програма спочатку імпортує необхідні бібліотеки:

- cv2 (OpenCV) для читання та запису відеофайлів;
- ruwt (PyWavelets) для виконання вейвлет-перетворень;
- numpy для операцій з масивами.

Функція вейвлет-фільтрації обробляє окремі кадри та відео обробляється окремо. Для цього кадр перетворюється з просторового домену в вейвлет-домен за допомогою дворівневого дискретного вейвлет-перетворення (`pywt.dwt2`).

Отримані вейвлет-коефіцієнти (LL, LH, HL, HH) модифікуються. Наприклад, можна встановити високочастотні коефіцієнти (LH, HL, HH) на нуль або зменшити їх величину, що допомагає у видаленні шуму.

Зворотне вейвлет-перетворення виконується через модифікації коефіцієнтів (`pywt.idwt2`), щоб перевести зображення назад у просторовий домен.

Читання та запис відео виконується наступним чином:

- відео читається кадр за кадром з використанням `cv2.VideoCapture`;
- до кожного кадра застосовується вейвлет-фільтрація;
- оброблені кадри записуються у новий відеофайл за допомогою `cv2.VideoWriter`;
- опційно, кадри можуть відображатися під час їх обробки.
- після обробки всіх кадрів, відеофайли (вхідний та вихідний) закриваються, і вікна `OpenCV` закриваються.

Треба пам'ятати, що вейвлет-фільтрація ефективно видаляє шум, зберігаючи при цьому важливі деталі зображення, завдяки своїй здатності розділити зображення на різні частотні компоненти. Ця методика особливо корисна для зображень зі складними текстурями, де інші методи фільтрації, такі як гауссовий або медіанний фільтр, можуть знищити важливі деталі зображення.

Видалення шуму з відео за допомогою методів машинного навчання стало значним напрямком у сфері обробки зображень та відео. Найчастіше це включає використання глибоких нейронних мереж, які навчаються на великих наборах даних для ідентифікації та видалення шуму з відео.

Розглянемо основні етапи роботи програми. Вибір та налаштування моделі використовують передтреновані моделі глибокого навчання, такі як DnCNN, UNet, або власні моделі, навчені на специфічних наборах даних.

Моделі можуть бути додатково налаштовані чи доналаштовані на конкретних наборах даних.

Обробка відео читається кадр за кадром. Кадри можуть бути перетворені або нормалізовані для відповідності вимогам моделі. Кожен кадр обробляється нейронною мережею для видалення шуму. Результат може вимагати додаткової обробки, наприклад, перетворення назад у відповідний формат. Це буде етапом постпроцесінгу. Запис обробленого відео записуються у новий відеофайл.

Фінальний етап роботи програми – закриваються всі файли та відповідні ресурси.

Реалізація може виглядати наступним чином. Створення програми для видалення шуму з відеорядів на основі машинного навчання вимагає використання бібліотеки глибокого навчання, такої як TensorFlow або PyTorch, та попередньо натренованої моделі. Одним із популярних підходів є використання конволюційних нейронних мереж (CNN), які можуть ефективно обробляти зображення та відео.

Розглянемо в деталях кроки для створення програми:

- вибір та імпортування бібліотек:

```
import cv2
```

```
import tensorflow as tf
```

- завантаження попередньо натренованої моделі:

Для цього прикладу, ми можемо використовувати модель, натреновану для видалення шуму. Модель можна завантажити з TensorFlow Hub або іншого джерела.

```
model = tf.keras.models.load_model('path_to_model')
```

Створюємо функцію для обробки кадрів:

```
def process_frame(frame, model):  
    # Перетворення кадру для моделі  
    input_frame = preprocess(frame)  
    # Обробка кадру моделлю  
    denoised_frame = model.predict(input_frame)  
    return postprocess(denoised_frame)
```

Додаємо читання та запис відео:

```
cap = cv2.VideoCapture('input_video.mp4')  
out = cv2.VideoWriter('output_video.mp4', cv2.VideoWriter_fourcc(*'MP4V'),  
30, (width, height))  
  
while cap.isOpened():  
    ret, frame = cap.read()  
    if not ret:  
        break  
    # Обробка кадру  
    denoised_frame = process_frame(frame, model)  
    # Запис обробленого кадру  
    out.write(denoised_frame)  
  
cap.release()  
out.release()
```

Необхідно пам'ятати, що `preprocess` і `postprocess` – це функції, які потрібно визначити для підготовки даних до введення в модель та обробки

виходу моделі відповідно. Перед використанням програми необхідно мати попередньо натреновану модель, яка спеціалізується на видаленні шуму.

Тепер, давайте розглянемо зображення, щоб продемонструвати можливий ефект видалення шуму з відеорядів за допомогою машинного навчання.



Рисунок 6.6 – Видалення шуму за допомогою машинного навчання

На зображенні ви можете побачити приклад використання машинного навчання для видалення шуму з фотографії міської нічної сцени. Ліва сторона зображення показує міську нічну сцену з видимим шумом, що має вигляд випадкової, зернистої текстури, представляючи стан «до» обробки. Права сторона зображення показує ту ж саму міську нічну сцену після застосування методів машинного навчання для видалення шуму, де шум значно зменшений, що призводить до чистішого та детальнішого вигляду, представляючи стан «після» обробки.

Після розгляду різних методів фільтрації для видалення шуму з відеоданих, можна зробити наступні висновки щодо вибору методів та їх застосування. Існує багато методів фільтрації, кожен з яких має свої переваги та недоліки. Вибір конкретного методу залежить від типу шуму, характеристик відеоданих та вимог до якості обробленого відео.

Гауссівський фільтр та медіанний фільтр є ефективними для загального шумоподавлення, але можуть призводити до розмиття деталей.

Білатеральний фільтр та фільтри на основі вейвлетів забезпечують краще збереження країв та деталей при видаленні шуму.

Методи на основі машинного навчання, зокрема глибокі нейронні мережі, відкривають нові можливості для більш ефективного видалення шуму. Вони можуть адаптуватися до різних типів шуму та забезпечувати вищу якість обробки.

Однак, вони вимагають значних обчислювальних ресурсів та великих обсягів даних для тренування. Є важливий компроміс між якістю видалення шуму та швидкістю обробки. Вища якість часто вимагає більше часу та обчислювальної потужності. Не існує універсального рішення для всіх сценаріїв. Вибір методу залежить від конкретного застосування, доступних ресурсів, та вимог до якості кінцевого продукту.

Можна підкреслити важливість ретельного вибору методу фільтрації з огляду на специфіку задачі та доступні ресурси, а також потенціал подальших досліджень у сфері використання машинного навчання для покращення якості видалення шуму. Дослідження методів видалення шуму з відеоданих є складною задачею, і ефективність вибору методу фільтрації значно визначається ретельністю підходу та адаптацією до конкретної задачі та умов дослідження. При виборі методу важливо враховувати кілька ключових аспектів.

По-перше, тип шуму грає велику роль у виборі методу. Різні алгоритми спрямовані на різні види шуму, тому необхідно чітко зрозуміти характер шуму у відеоданих.

Другим важливим фактором є обчислювальна складність обраного методу. Враховуючи доступні ресурси, слід обрати той метод, який може бути використаний у відповідних обсягах часу та обчислювальних потужностей.

Третій аспект – втрати якості. Деякі методи можуть призводити до втрати деталей та якості відео, тому важливо зважити на прийнятність цих втрат в контексті конкретного дослідження.

Також, враховуючи сучасні технології, використання методів машинного навчання для видалення шуму може бути ефективним підходом, особливо якщо доступні великі обсяги відмічених даних.

Адаптивність методу до різних умов зйомки та доступність реалізацій також грають важливу роль у виборі методу.

Загалом, оптимальний метод видалення шуму буде результатом глибокого аналізу та порівняння різних підходів, приділяючи увагу вищезазначеним факторам, щоб забезпечити ефективність та прийнятність результатів дослідження.

ВИСНОВКИ

У рамках дипломної кваліфікаційної роботи з теми «Дослідження методів видалення шуму з відеоданих» було проведено глибокий теоретичний аналіз різних джерел шуму в відеоданих та методів їх видалення. В ході роботи було розглянуто як традиційні методи фільтрації, такі як медіанний фільтр, так і передові підходи, засновані на машинному навчанні, включаючи конволюційні нейронні мережі та генеративно-змагальні мережі.

У практичній частині роботи було реалізовано кілька алгоритмів видалення шуму. Результати застосування цих методів продемонстрували їх високу ефективність у покращенні якості зображень. Представлені зразки зображень до та після обробки чітко вказували на значне зниження рівня шуму, при цьому не втрачаючи важливих деталей зображення.

Робота підкреслила значний потенціал машинного навчання у вирішенні проблеми шуму в відеоданих. Зокрема, глибоке навчання відкриває нові можливості для створення більш точних та адаптивних систем видалення шуму. Також було виявлено, що існує потреба у подальших дослідженнях, наприклад, у розробці гібридних моделей, які поєднують класичні алгоритми та методи машинного навчання, а також у створенні спеціалізованих рішень для обробки відео в конкретних умовах.

Результати цієї роботи мають значний практичний інтерес для розвитку технологій обробки відеоданих. Це особливо важливо у сферах відеонагляду, кінематографії та обробки медичних зображень, де якість зображень має критичне значення. В цілому, виконана дипломна робота вносить важливий вклад у дослідження методів видалення шуму з відеоданих, відкриваючи нові шляхи для подальших досліджень та розробок у цій області.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Buades, A., Coll, B., & Morel, J. M. (2005). A non-local algorithm for image denoising. In CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Vol. 2, pp. 60-65). IEEE.
2. Chang, H., Yeung, D. Y., & Xiong, Y. (2004). Super-resolution through neighbor embedding. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (Vol. 1, pp. I-275). IEEE.
3. Chen, S., Zhou, Y., Zhang, L., Tian, Q., & Yao, H. (2015). Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1693-1701). IEEE.
4. Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8), 2080-2095.
5. Dabov, K., Foi, A., & Katkovnik, V. (2010). Image denoising by sparse 3-D transform-domain collaborative filtering: Theoretical analysis and implementation. *Journal of the Optical Society of America A*, 27(12), 2524-2536.
6. Damera-Venkata, N., & Nowak, R. D. (2003). Nonlinear denoising of images with edge detection constraints. *IEEE Transactions on Image Processing*, 12(5), 533-545.
7. Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12), 3736-3745.
8. Foi, A., Katkovnik, V., & Egiazarian, K. (2008). Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Transactions on Image Processing*, 17(5), 539-550.

9. Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (3rd ed.). Pearson Education.
10. Liu, C., Szeliski, R., Kang, S. B., Zitnick, C. L., & Freeman, W. T. (2008). Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 299-314.
11. Portilla, J., & Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1), 49-71.
12. Portilla, J., Strela, V., Wainwright, M. J., & Simoncelli, E. P. (2003). Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11), 1338-1351.
13. Pratt, W. K. (2007). *Digital Image Processing: PIKS Inside*. John Wiley & Sons.
14. Rasti, P., & Milanfar, P. (2012). Image denoising in mixed Poisson-Gaussian noise. *IEEE Transactions on Image Processing*, 21(2), 858-868.
15. Roth, S., & Black, M. J. (2005). Fields of experts: A framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 2, pp. 860-867). IEEE.
16. Timofte, R., De Vleeschouwer, C., & Van Gool, L. (2013). Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 1920-1927). IEEE.
17. Timofte, R., Smet, V., Rothe, R., & Van Gool, L. (2016). Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 281-289). IEEE.
18. Zhang, K., Gao, X., Tao, D., & Li, X. (2013). Detail-revealing adaptive non-local means for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 1050-1057). IEEE.

19. Zhang, K., Tao, D., Gao, X., & Li, X. (2010). Multi-layer group sparsity of LR and HR patches for scalable face hallucination. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 356-369). Springer.

20. Zoran, D., & Weiss, Y. (2011). From learning models of natural image patches to whole image restoration. In 2011 International Conference on Computer Vision (ICCV) (pp. 479-486). IEEE.

21. Gadetska, S. V., Gorokhovatskyi, V. O., Stiahlyk, N. I., & Vlasenko, N. V. (2021). Statistical data analysis tools in image classification methods based on the description as a set of binary descriptors of key points. Radio Electronics, Computer Science, Control, (4), 58-68.

22. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, Міжн. наук. симпозіум Інтелектуальні рішення-С. Обчислювальний інтелект. Теорія прийняття рішень: праці міжн. наук. симп. (Вересень 29, 2021). Київ – Ужгород, С. 44-45.

23. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, Сучасні інформаційні системи, 7(1), С. 5-13.

24. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, Сучасні інформаційні системи, 6(3), С. 5-12.

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

26. Гороховатский В.А., Передрий Е.О. (2009) Кореляційні методи розпізнавання зображень шляхом голосування систем фрагментів. Радіоелектроніка, інформатика, управління, №1 (20), с.74-81.

27. Gadetska S., Gorokhovatskyi V., Stiahlyk N., Vlasenko N. (2022) Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2022), 3137, pp. 68-77.

28. Гороховатський, В. О., & Власенко, Н. В. (2021). Редукція опису зображення у складі множини дескрипторів на основі метричного критерію інформативності.

29. Gadetska S., Gorokhovatskyi V., Stiahlyk N., Vlasenko N. (2022) Handwritten character recognition models based on convolutional neural networks, International Journal of Academic Engineering Research, 7(9), 64-72.

30. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.

31. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), pp. 57-70.