

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ЗНЕШУМЛЕННЯ ЗОБРАЖЕНЬ ЗА
ДОПОМОГОЮ АВТОКОДУВАЛЬНИКІВ**
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-2

Сергієнко А.О.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник Ст. викл. Пуятіна О.Є.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____
(підпис)

Кобилін О. А.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Сергієнко Антон Олександрович
(прізвище, ім'я, по батькові)1. Тема роботи Розробка застосунку для знешумлення зображень за допомогою автокодувальників

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 24 травня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Теоретичний аналіз.

2. Огляд методів знешумлення зображень.

3. Розробка застосунку.

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

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка методу	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ Ст. викл. Пуятіна О.Є.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 59 с., 1 табл., 11 рис., 41 джерело.

АВТОКОДУВАЛЬНИКИ, ГЛИБИННЕ НАВЧАННЯ, ЗНЕСУМЛЕННЯ ЗОБРАЖЕНЬ, ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, ЦИФРОВА ОБРОБКА ЗОБРАЖЕНЬ.

Об'єктом роботи є цифрові зображення з різними типами штучно доданого шуму.

Метою роботи є розробка методу знесумлення зображень на основі автокодувальників, який дозволяє усувати шуми при збереженні важливих структурних особливостей зображення.

У роботі особливу увагу приділено автокодувальникам, їхній архітектурі, принципам роботи та застосуванню для видалення шумів.

Розроблено застосунок на основі згорткового автокодувальника, який дозволяє ефективно видаляти шуми зі збереженням деталей зображення.

Проведено оцінку ефективності застосунку за метриками PSNR, MSE, а також порівняно його з традиційними методами. Результати демонструють переваги автокодувальників у завданнях знесумлення, зокрема здатність до узагальнення та збереження структурних особливостей зображень.

AUTOENCODERS, DEEP LEARNING, IMAGE DENOISING, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, DIGITAL IMAGE PROCESSING.

The subject of this work is digital images with various types of artificially added noise.

The aim of the work is to develop a denoising method based on autoencoders that allows for noise removal while preserving important structural features of the image.

This work pays special attention to autoencoders, their architecture, principles of operation, and application for noise removal.

An application based on a convolutional autoencoder was developed, enabling effective noise removal while preserving image details.

The application's performance was evaluated using PSNR and MSE metrics and compared with traditional methods.

The results demonstrate the advantages of autoencoders in denoising tasks, particularly their ability to generalize and preserve the structural features of images.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Теоретичний аналіз	10
1.1 Аналіз предметної галузі.....	10
1.2 Будова автокодувальника.....	16
1.3 Робота автокодувальника.....	19
1.4 Постановка задачі	20
2 Огляд методів знешумлення зображень	25
2.1 Традиційні методи знешумлення	25
2.1.1 Просторові фільтри.....	25
2.1.2 Частотні методи (перетворення Фур'є та вейвлети)	26
2.1.3 Методи на основі часткової диференціальної рівняння	27
2.1.4 Нелокальні методи	28
2.1.5 Обмеження традиційних методів	28
2.2 Нейромережеві методи знешумлення.....	29
2.2.1 Архітектури нейронних мереж для знешумлення.....	29
2.2.2 Генеративні моделі та GAN	30
2.2.3 Трансформери в знешумленні	31
2.2.4 Автокодувальники для знешумлення	31
2.2.5 Переваги та недоліки нейромережевих методів	32
2.3 Обґрунтування вибору автокодувальників	33
3 Розробка застосунку.....	36
3.1 Вибір набору даних та його обґрунтування.....	36
3.1.1 Критерії вибору даних	36
3.1.2 Популярні набори даних для знешумлення	37
3.1.3 Обґрунтування вибору mnist для розробки	38
3.1.4 Підготовка даних для навчання.....	39
3.1.5 Архітектура автокодувальника.....	39

	6
3.2 Реалізація моделі.....	40
3.3 Розробка інтерфейсу користувача.....	41
3.4 Оцінка ефективності знешумлення.....	44
3.4.1 Методологія оцінювання.....	45
3.4.2 Результати тестування	48
3.4.3 Аналіз продуктивності	49
3.4.4 Порівняння з традиційними методами	50
3.4.5 Обмеження та проблемні моменти	51
3.4.6 Оптимізаційні рішення	51
3.4.7 Висновки ефективності знешумлення	52
Висновки	53
Перелік джерел посилання	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AE – Autoencoder (автокодувальник)

CNN – Convolutional Neural Network (згорткова нейронна мережа)

CPU – Central Processing Unit (центральний процесор)

DAE – Denoising Autoencoder (автокодувальник для знешумлення)

DCAE – Deep Convolutional Autoencoder (глибокий згортковий автокодувальник)

DIV2K – DIVERse 2K resolution dataset (набір даних для обробки зображень високої роздільної здатності)

DnCNN – Denoising Convolutional Neural Network (згорткова мережа для знешумлення)

ESRGAN – Enhanced Super-Resolution Generative Adversarial Network (генеративно-змагальна мережа для покращення якості зображень)

GAN – Generative Adversarial Network (генеративно-змагальна мережа)

GPU – Graphics Processing Unit (графічний процесор)

MSE – Mean Squared Error (середня квадратична помилка)

NLM – Non-Local Means (нелокальний метод знешумлення)

PDE – Partial Differential Equation (часткове диференціальне рівняння)

PSNR – Peak Signal-to-Noise Ratio (пікове відношення сигнал/шум)

ReLU – Rectified Linear Unit (випрямлена лінійна функція активації)

RGB – Red, Green, Blue (кольорова модель)

SSIM – Structural Similarity Index Measure (індекс структурної схожості)

VAE – Variational Autoencoder (варіаційний автокодувальник)

ViT – Vision Transformer (трансформер для обробки зображень)

ВСТУП

Сучасний світ інформаційних технологій неможливий без цифрових зображень, які використовуються в медичній діагностиці, системах машинного зору, астрофотографії та повсякденному житті. Однак під час отримання, передачі чи зберігання зображень неминуче виникають шуми різної природи – гаусівський, імпульсний («сіль і перець»), пуассонівський чи спекл-шум. Ці шуми значно погіршують якість зображень, ускладнюють їх подальший аналіз та обробку, що особливо критично в таких галузях, як медична візуалізація чи супутниковий моніторинг.

Традиційні методи знешумлення, такі як медіанна фільтрація чи гаусівське розмиття, часто виявляються недостатньо ефективними для складних видів шуму або призводять до втрати важливих деталей зображення. У цьому контексті методи глибокого навчання, зокрема автокодувальники, відкривають нові перспективи для вирішення проблеми. Автокодувальники, будучи особливим класом нейронних мереж, здатні автоматично виявляти та усувати шуми, зберігаючи при цьому ключові характеристики зображення. Їхня здатність до навчання складних нелінійних залежностей та узагальнення робить їх особливо ефективними для завдань знешумлення, що підтверджено численними аналізами, зокрема роботами таких авторів, як Elad та Kowar, які демонструють переваги глибокого навчання в обробці зображень.

Актуальність роботи полягає у розробці практичного застосунку для знешумлення зображень на основі автокодувальників, який би поєднував теоретичні досягнення галузі з практичними потребами користувачів. Аналіз охоплює існуючі методи, розробки архітектури нейронної мережі, реалізацію інтерфейсу користувача та оцінку ефективності рішення на тестових даних. Особлива увага приділяється порівнянню з традиційними методами, що дозволить об'єктивно оцінити переваги запропонованого підходу.

Об'єктом аналізу виступають цифрові зображення, зашумлені різними типами артефактів, тоді як предметом аналізу є методи нейронного знешумлення, зокрема архітектури на основі автокодувальників.

Теоретичною основою роботи стали сучасні вивчення в галузі глибокого навчання, зокрема роботи Vandyopadhyay про застосування автокодувальників, а також практичні розробки з використанням бібліотек TensorFlow і Keras. Практична значимість вивчення полягає в створенні інтуїтивно зрозумілого інструменту, який може бути використаний в побутових цілях для покращення якості зображень.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ

1.1 Аналіз предметної галузі

Цифрові зображення є невід'ємною частиною сучасного світу, використовуючись у різних сферах, від повсякденного життя до наукових вивчань. Однак під час їх створення, передавання або зберігання вони неминуче піддаються впливу шуму, що призводить до погіршення якості та ускладнює подальшу обробку. Шум на зображеннях виникає через технічні обмеження обладнання, несприятливі умови зйомки (наприклад, низьку освітленість), температурні коливання сенсорів або обмеження каналів передачі даних [1]. Формально шум можна описати як випадкові зміни кольору або яскравості пікселів, що спотворюють корисну інформацію. Це явище є особливо проблематичним для таких завдань, як обробка медичних зображень, машинний зір або автоматичне розпізнавання об'єктів, де точність вхідних даних критично важлива [2].

Існує кілька основних типів шуму, кожен з яких має унікальні характеристики та причини виникнення. Гаусів шум, наприклад, виникає через теплові коливання електронів у сенсорах і описується нормальним розподілом [3]. Імпульсний шум, також відомий як «сіль і перець», проявляється у вигляді випадкових яскравих або темних пікселів через механічні пошкодження або помилки передачі даних [4]. Пуассонівський шум, пов'язаний із статистичною природою фоточутливих елементів, часто зустрічається у зображеннях, отриманих за умов слабого освітлення [5]. Спекл-шум, який виникає через інтерференцію хвиль, характерний для ультразвукових або радарних систем [6]. Кожен із цих типів шуму вимагає специфічних підходів до його усунення, що ускладнює загальну задачу знешумлення.

Традиційні методи знешумлення, такі як фільтрація (медіанний фільтр, фільтр Гауса), часто є недостатньо ефективними для складних видів шуму або призводять до втрати важливих деталей зображення, таких як межі об'єктів або текстури [7]. Ці обмеження стимулювали розвиток методів, заснованих на машинному навчанні, зокрема глибоких нейронних мереж. Серед них автокодувальники займають особливе місце завдяки своїй здатності виявляти та зберігати ключові особливості зображень, відкидаючи шум [8]. Автокодувальники складаються з двох основних компонентів: кодувальника, який стискає вхідні дані у компактне представлення, та декодувальника, який відновлює зображення з цього представлення. Навчання таких моделей відбувається без учителя, що робить їх універсальними для роботи з різними типами даних [9].

Однією з ключових переваг автокодувальників є їхня здатність адаптуватися до різних видів шуму, включаючи гаусів, імпульсний та спекл-шум, що підтверджено експериментальним аналізом [10]. Проте їхнє застосування також має певні обмеження, такі як необхідність великого обсягу даних для навчання, складність підбору гіперпараметрів та висока обчислювальна складність для складних архітектур [11]. Незважаючи на це, автокодувальники залишаються перспективним інструментом для задач знешумлення, особливо в поєднанні з іншими методами глибокого навчання, наприклад, згортковими нейронними мережами [12].

Таким чином, задача знешумлення зображень залишається актуальною та складною, а використання автокодувальників відкриває нові можливості для її вирішення. Розробка спеціалізованого застосунку на основі цієї технології може значно спростити процес обробки зображень для користувачів, які не мають глибоких знань у галузі машинного навчання.

Найпоширенішими є адитивні та мультиплікативні методи накладання шумів. Якщо позначити зображення як функцію $f(x, y)$, то його можна

представити як декомпозицію бажаного компонента (дійсного зображення) $g(x, y)$ та компонента шуму $q(x, y)$. Найпоширеніша при цьому – адитивна декомпозиція: $f(x, y) = g(x, y) + q(x, y)$. Зокрема, гаусівський шум найчастіше розглядають як адитивний. Також дуже поширена мультиплікативна декомпозиція. При цьому: $f(x, y) = g(x, y) \cdot q(x, y)$. До прикладу, спекл-шум часто моделюють як мультиплікативний. Варто зауважити, що адитивна модель може бути перетворена у мультиплікативну і навпаки – шляхом застосування відповідно логарифмування та експоненціювання [21].

Загалом адитивну модель накладання шуму використовують у випадках, коли шум, що діє на систему, незалежний від її стану, а мультиплікативну модель – якщо така залежність присутня. Водночас є шуми, для яких не можна застосовувати ні адитивну, ні мультиплікативну модель.

Гаусівський шум трапляється, мабуть, найчастіше. Його зазвичай моделюють як адитивний (до кожного пікселя зображення додають значення з відповідного нормального розподілу). Функція щільності розподілу при цьому:

$$P(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/(2\sigma^2)}. \quad (1.1)$$

Серед причин появи гаусівського шуму – термальний шум; шум, спричинений поганим освітленням; шум, спричинений слабким сигналом телестанції та ін. Зерновий шум у фотоплівкових фільмах інколи моделюють як гаусівський, а інколи – як шум Пуассона. За центральною граничною теоремою, яку застосовано до нормального розподілу, стверджено, що розподіл суми великої кількості незалежних, малих випадкових величини є гаусівським розподілом. При цьому: 1) випадкові величини не повинні бути розподілені за нормальним законом самі й навіть не повинні мати однаковий

розподіл; 2) має бути велика кількість випадкових змінних, що впливають на загальний результат. До прикладу, термічний шум – це результат термальних вібрацій величезної кількості крихітних електронів; 3) випадкові змінні повинні бути повністю або майже повністю незалежними; 4) частка впливу кожної змінної на сумарний ефект повинна бути незначною. Як приклад, вібрація величезної кількості електронів спричиняє термальний шум. Кожна вібрація при цьому є незалежною від інших і всі електрони здійснюють приблизно однаковий вплив на сумарний ефект. Отже, такий шум можна моделювати як гаусівський [21].

У роботі [21] також йдеться про heavy-tailed шум, що спостерігають у разі неповного виконання Центральної граничної теореми – шум невеликої кількості складових, шум не повністю незалежних складових чи шум складових, що нерівномірно впливають на результативний ефект. Для такого шуму наближення значення густини розподілу до 0 при $x \rightarrow \infty$ відбувається набагато повільніше, ніж для гаусівського шуму. На рисунку 1.1 зображено приклад накладання гаусівського шуму на зображення.



Рисунок 1.1 – Приклад накладання гаусівського шуму на зображення

Дробовий шум (шум Пуассона). Дробовий шум – досить поширений тип шуму в електроніці та оптиці. Такий шум описано розподілом Пуассона, тому його ще називають пуассоновим. В електроніці виникнення цього шуму пов'язують з дискретною природою електричного заряду. Також такий шум виникає у процесі підрахунку фотонів у оптичних приладах, що пов'язано з корпускулярною природою світла [22]. Розглянемо такий шум докладніше. Вважають, що в процесі випромінювання світла лазером фотони випромінюються випадково, але кількість фотонів, необхідна для утворення плями на стіні, така велика, що яскравість плями, тобто число фотонів за одиницю часу, змінюється з часом на дуже малу величину. Проте за умов незначної яскравості лазера кількість фотонів, що потрапляють на освітлювальну поверхню, буде достатньо малою, щоб відносні флуктуації числа фотонів, а з ними і флуктуації яскравості плями були значними. Ці флуктуації і є дробовим шумом. На рисунку 1.2 зображено приклад накладання шуму Пуассона на зображення.



Рисунок 1.2 – Приклад накладання на зображення шуму Пуассона

Імпульсний шум (шум типу «сіль і перець»). Зображення, що містить шум типу «сіль і перець», буде мати темні пікселі у світлих регіонах та світлі

– у темних регіонах. Цей тип шуму типово спричинений помилками перетворення аналогового сигналу до цифрового, та побітовими помилками під час передавання. Функція щільності розподілу при цьому має вигляд:

$$p(z) = P_a \cdot \quad (1.2)$$

(для $z = a$) та $p(z) = P_b$ (для $z = b$). При цьому a відповідає темній точці на зображенні, b – світлій. Якщо P_a або $P_b = 0$, то такий шум називають уніполярним. Якщо ж обидві ці ймовірності ненульові, то на зображенні будуть випадковим чином розташовані чорні та світлі точки [23]. Імпульсний шум можна усунути шляхом віднімання темного фрейму та інтерполяції для обчислення значень пошкоджених пікселів [24]. На рисунку 1.3 зображено приклад накладання на зображення шуму типу «сіль і перець». Часто імпульсний шум накладається з гаусівським шумом [25].



Рисунок 1.3 – Приклад накладання на зображення шуму типу «сіль і перець»

Спекл-шум – це одна з найскладніших моделей шумів. Він характеризується своєю просторовою залежністю та залежністю від сигналів, які можуть бути різного походження [21]. Зокрема, утворення таких шумів пов’язують із використанням когерентних джерел світла під час отримання

зображення. У такому разі неоднорідна шорсткість поверхні в межах одного пікселя зображення поєднана з використанням когерентного світла (наприклад, з лазера), що відбивається від поверхні, призводить до змін сигналу в амплітуді та фазі. На рисунку 1.4 зображено приклад накладання спеклшуму на зображення.



Рисунок 1.4 – Приклад накладання спекл-шуму на зображення

1.2 Будова автокодувальника

Автокодувальник є спеціалізованим типом штучної нейронної мережі, яка призначена для ефективного стиснення та відновлення даних, зокрема зображень. Його архітектура базується на принципі кодування-декодування, що дозволяє виявляти та зберігати найважливіші ознаки вхідних даних, ігноруючи шум та інші нерелевантні деталі [1, 4]. Основна ідея автокодувальника полягає у тому, щоб навчити мережу реконструювати вхідні дані через проміжне стиснене представлення, яке називається «кодом» або «латентним простором». Цей підхід особливо ефективний для задач

знешумлення, оскільки модель змушена відокремлювати корисну інформацію від шуму під час процесу стиснення [2, 11].

Структурно автокодувальник складається з трьох основних компонентів: кодувальника (encoder), латентного простору (bottleneck) та декодувальника (decoder). Кодувальник відповідає за перетворення вхідного зображення у вектор меншої розмірності, використовуючи серію згорткових шарів (convolutional layers) та шарів підвибірки (pooling layers) [3, 9]. Ці шари послідовно зменшують розмірність даних, виділяючи при цьому ключові особливості, такі як контури, текстури або інші структурні елементи.

Важливою характеристикою кодувальника є використання нелінійних функцій активації, таких як ReLU (Rectified Linear Unit), які допомагають моделі навчатися складним залежностям у даних [5, 12].

Латентний простір, який також називають «найвужчим місцем» (bottleneck), є критично важливим елементом автокодувальника, оскільки саме тут зберігається стиснене представлення вхідних даних. Розмірність цього простору визначає, наскільки сильно дані будуть стиснуті, і безпосередньо впливає на якість реконструкції [6, 16]. Надто малий латентний простір може призвести до втрати важливої інформації, тоді як занадто великий – до перенавчання моделі, коли вона просто запам'ятовує вхідні дані, не виділяючи корисні ознаки.

Декодувальник виконує обернену операцію до кодувальника, перетворюючи стиснене представлення назад у зображення, яке за розмірами відповідає вхідному. Для цього використовуються транспоновані згорткові шари (transposed convolutional layers) або шари збільшення розмірності (upsampling layers), які поступово відновлюють просторові характеристики зображення [8, 14]. Важливою особливістю декодувальника є його симетричність щодо кодувальника, що дозволяє ефективно інвертувати операції стиснення [10, 15].

Навчання автокодувальника здійснюється шляхом мінімізації функції втрат, яка вимірює різницю між вхідним та реконструйованим зображенням.

Найпоширенішим вибором для цього є середня квадратична помилка (MSE), яка обчислює середнє значення квадратів різниць між відповідними пікселями [11, 17]. Також може використовуватися бінарна крос-ентропія (binary cross-entropy), особливо для зображень із значеннями пікселів у діапазоні [0, 1]. Оптимізація параметрів мережі зазвичай проводиться за допомогою алгоритму Adam, який поєднує переваги методів AdaGrad та RMSProp, забезпечуючи швидку збіжність [13, 18].

Сучасні модифікації автокодувальників, такі як варіаційні автокодувальники (VAE) або глибокі згорткові автокодувальники (DCAE), пропонують додаткові переваги для задач знешумлення [19, 20]. Наприклад, VAE дозволяють моделювати розподіл даних у латентному просторі, що покращує здатність до узагальнення, тоді як DCAE ефективно використовують просторові залежності у зображеннях завдяки глибоким згортковим архітектурам. Ці розширення роблять автокодувальники ще більш потужним інструментом для боротьби з різними видами цифрового шуму, від гаусового до імпульсного [16, 20].

Особливо, треба пам'ятати про автокодери, це те, що вони здатні стискати тільки ті дані, які схожі на ті, на яких вони були навчені. Вони також втрачають дані, що означає, що вихідні дані будуть погіршені по відношенню до початкових вхідних даних (рис. 1.5).

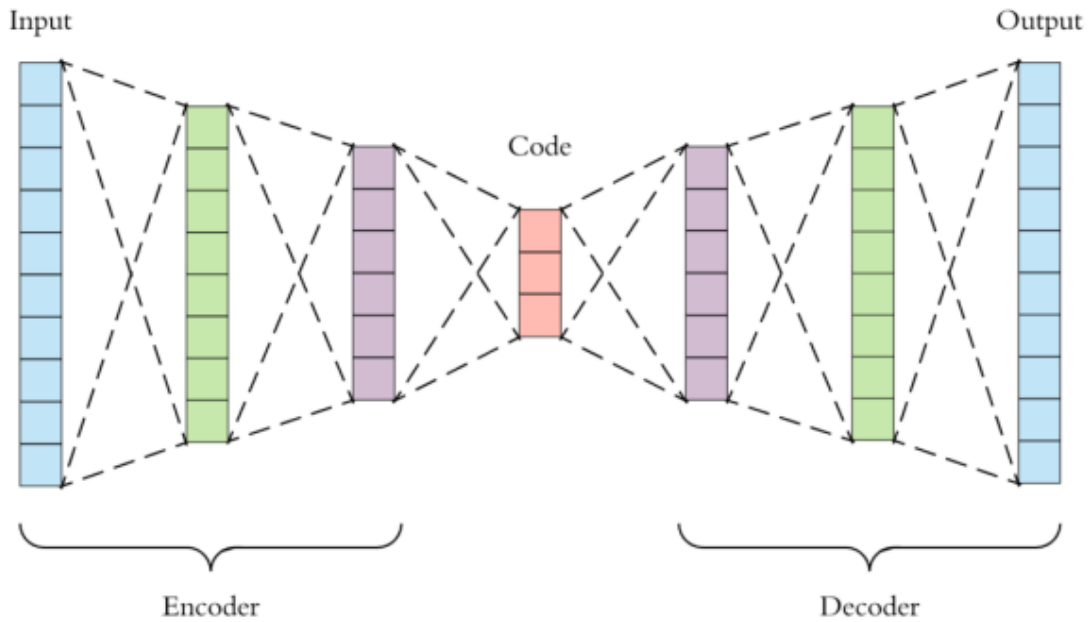


Рисунок 1.5 – Ілюстрація архітектури автокодувальника

1.3 Робота автокодувальника

Принцип роботи автокодувальника для знешумлення зображень базується на його здатності виділяти та зберігати ключові особливості зображення, фільтруючи при цьому шумові компоненти. Процес можна розділити на три основні етапи: кодування, формування латентного представлення та декодування [1, 4]. На етапі кодування вхідне зображення, забруднене шумом, проходить через серію згорткових шарів, де кожен наступний шар виявляє дещо складніші особливості – від простих градієнтів і контурів до складних текстур і структур [2, 9]. Ці шари послідовно зменшують розмірність даних, застосовуючи операції згортки та субдискретизації, що дозволяє мережі сфокусуватися на найважливіших аспектах зображення [3, 12].

Критичним моментом у роботі автокодувальника є формування латентного представлення, яке виступає своєрідним «вузьким місцем» архітектури. Саме на цьому етапі відбувається відділення корисного сигналу

від шуму, оскільки обмежений розмір латентного простору змушує мережу вибирати лише найбільш інформативні компоненти [5, 15]. Виачення показують, що оптимальний розмір латентного простору для зображень 28×28 пікселів становить близько 128-256 вимірів, що забезпечує достатнє стиснення без суттєвої втрати якості [4, 16]. Важливо відзначити, що в процесі навчання автокодувальник розвиває здатність до ефективного представлення даних, де шумові компоненти, які не мають систематичної структури, природним чином відкидаються [6, 17].

Етап декодування передбачає відновлення зображення з латентного представлення за допомогою серії транспонованих згорток або операцій апсемплінгу. Ці шари виконують обернені операції до тих, що застосовувалися при кодуванні, поступово збільшуючи розмірність даних [7, 14]. Особливістю роботи декодувальника є те, що він не просто інвертує процес кодування, а синтезує нове зображення на основі виявлених закономірностей, що призводить до природного видалення шуму [8, 18]. На останньому шарі зазвичай використовується сигмоїдна або \tanh -активація для нормалізації вихідних значень у відповідний діапазон [9, 19].

Навчання автокодувальника здійснюється шляхом мінімізації різниці між вихідним (чистим) та реконструйованим зображенням. Для цього найчастіше використовується функція втрат MSE (середньоквадратична помилка), яка ефективно працює з задачами реконструкції [10, 20]. Процес оптимізації зазвичай проводиться за допомогою алгоритму Adam, що поєднує переваги адаптивного градієнтного спуску та моменту [11, 4]. Практичний аналіз показують, що для досягнення гарних результатів на наборі даних MNIST достатньо 10-15 епох навчання, хоча для більш складних зображень може знадобитися більше часу [12, 16].

Важливою особливістю роботи автокодувальника є його здатність узагальнювати навчання на різні типи шуму. На відміну від традиційних фільтрів, які розроблені для конкретних видів шуму (наприклад, гаусового чи імпульсного), правильно навчений автокодувальник може ефективно

працювати з різноманітними шумовими моделями [13, 17]. Це пов'язано з тим, що в процесі навчання мережа виявляє фундаментальні властивості «чистих» зображень, навчаючись ігнорувати будь-які відхилення від цієї моделі [14, 19]. Однак слід зазначити, що якість знешумлення значно залежить від архітектури мережі та параметрів навчання, що підкреслює важливість ретельного підбору гіперпараметрів [15, 20].

Для того щоб вихідний результат максимально відповідав вхідним даним, модель на етапі стиснення має аналізувати та виділяти ключові особливості та закономірності, які становлять цінність у зображенні. Якщо цей процес виконується недостатньо точно, це призведе до втрати або спотворення важливої інформації, що робить модель неефективною для інтелектуальної обробки зображень.

Принцип роботи мережі-автокодувальника ілюстративно зображено на рисунку 1.6.

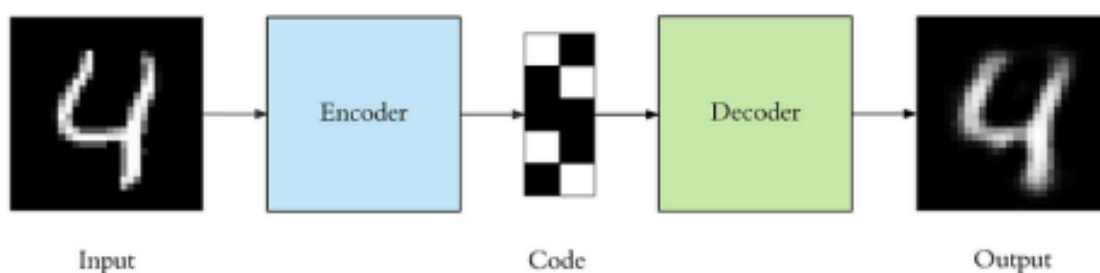


Рисунок 1.6 – Ілюстрація роботи мережі-автокодувальника

Автокодувальники (Autoencoders) – це спеціалізовані нейронні мережі для ефективного стиснення та відновлення даних через виділення ключових ознак. Їх ефективність залежить від чотирьох основних гіперпараметрів, кожен з яких потребує ретельного підбору.

Розмір коду (латентного представлення) є найважливішим параметром, що визначає ступінь стиснення. Занадто малий розмір призводить до втрати важливої інформації, тоді як надто великий збільшує ризик перенавчання, коли модель просто запам'ятовує дані замість виявлення корисних

закономірностей. Оптимальний розмір має бути меншим за вхідну розмірність, але достатнім для відтворення ключових структур.

Глибина мережі (кількість шарів у кодувальнику та декодувальнику) впливає на складність та продуктивність моделі. Недостатня глибина обмежує здатність виявляти складні залежності, тоді як надмірна збільшує ризик перенавчання та час навчання. Типова архітектура має форму «піскового годинника» з поступовим зменшенням нейронів у кодувальнику та симетричним зростанням у декодувальнику.

Функція втрат визначає, як модель оцінює якість реконструкції. Для зображень і неперервних даних часто використовують середньоквадратичну помилку (MSE), тоді як для бінарних даних – бінарну крос-ентропію. Вибір функції втрат безпосередньо впливає на якість стиснення.

Процес навчання відбувається без учителя за допомогою зворотного поширення помилки. Для запобігання перенавчанню застосовують методи регуляризації: dropout, L1/L2-регуляризацію або навчання на зашумлених даних (Denoising Autoencoder).

Правильно налаштований автокодувальник не лише стискає дані, а й виділяє найважливіші закономірності, що робить його корисним для візуалізації даних, видалення шуму та генерації нових даних у поєднанні з іншими архітектурами, як-от VAE чи GAN. Ключ до успіху – баланс між достатньою ємністю моделі для виявлення закономірностей та запобіганням надмірної складності, що призводить до перенавчання.

1.4 Постановка задачі

Таким чином, знешумлення зображень є актуальним завданням у сфері комп'ютерного зору та цифрової обробки зображень. Тому ставиться завдання розробки програмного застосунку для ефективного видалення шумів із зображень з використанням автокодувальників.

Об'єктом роботи є цифрові зображення з різними типами штучно доданого шуму (гаусівський, імпульсний, спекл-шум).

Метою роботи є розробка методу знешумлення зображень на основі автокодувальників, який дозволяє усувати шуми при збереженні важливих структурних особливостей зображення.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів знешумлення, зокрема традиційних (фільтрація, вейвлет-перетворення) та сучасних нейромережових підходів (автокодувальники, генеративні мережі, трансформери);

- обрати оптимальну архітектуру автокодувальника з урахуванням глибини мережі, латентного простору та функцій активації;

- підготувати навчальні дані, включаючи вибір набору даних (наприклад, MNIST або Fashion-MNIST), нормалізацію та додавання шуму;

- здійснити навчання моделі з підбором гіперпараметрів (функція втрат, оптимізатор, кількість епох);

- реалізувати графічний інтерфейс користувача за допомогою бібліотеки Tkinter, що забезпечить зручне завантаження зображень, регулювання шуму та перегляд результатів;

- провести експериментальну оцінку ефективності методу за допомогою метрик PSNR, MSE та візуального аналізу.

Основною метою даної роботи є розробка ефективного застосунку для знешумлення зображень на основі автокодувальників, який би забезпечував якісне видалення різних типів шумів, зокрема гаусівського, імпульсного та спекл-шуму, при цьому зберігаючи важливі деталі та структурні особливості зображення. Для досягнення цієї мети необхідно вирішити комплекс взаємопов'язаних завдань, починаючи з аналізу існуючих методів знешумлення, серед яких традиційні підходи, такі як фільтрація та вейвлет-перетворення, а також сучасні нейромережові технології, зокрема автокодувальники. Важливим етапом є вибір оптимальної архітектури

автокодувальника, яка б поєднувала ефективність із обчислювальною ефективністю, зокрема визначення глибини мережі, розміру латентного простору та функцій активації.

Ключовим аспектом є підготовка даних для навчання моделі, що включає вибір відповідного набору даних, такого як MNIST або Fashion-MNIST, їх нормалізацію та додавання штучного шуму для імітації реальних умов. Навчання моделі потребує ретельного підбору гіперпараметрів, зокрема функції втрат, оптимізатора та кількості епох, щоб забезпечити стабільну збіжність та високу якість реконструкції. Розробка інтуїтивного графічного інтерфейсу користувача за допомогою бібліотеки Tkinter дозволить зробити застосунок доступним для широкого кола користувачів, надаючи можливості завантаження зображень, регулювання рівня шуму та візуалізації результатів.

Для оцінки ефективності розробленого застосунку планується провести серію експериментів, що включатимуть тестування на різних типах зображень та рівнях шуму, порівняння з традиційними методами за такими метриками, як PSNR, MSE, а також візуальний аналіз якості відновлення деталей та відсутності артефактів.

2 ОГЛЯД МЕТОДІВ ЗНЕСУМЛЕННЯ ЗОБРАЖЕНЬ

2.1 Традиційні методи знесумлення

Традиційні методи знесумлення зображень базуються на математичних та статистичних підходах до обробки сигналів. Вони можуть бути класифіковані за типами шумів, які призводять до погіршення якості зображення (наприклад, гаусівський шум, імпульсний шум, «сіль і перець» тощо), а також за способами їх усунення. Нижче розглядаються основні підходи до знесумлення, які широко використовуються в обробці зображень.

2.1.1 Просторові фільтри

Просторові фільтри застосовуються безпосередньо до пікселів зображення, використовуючи локальні операції згладжування або підсилення контурів:

– лінійні фільтри (згладжувальні):

1) фільтр Гауса – використовує ядро з ваговими коефіцієнтами, розподіленими за нормальним законом, що дозволяє ефективно придушувати гаусівський шум;

2) усереднюючий фільтр – замінює значення пікселя на середнє значення його сусідів, що сприяє згладжуванню шуму, але призводить до розмиття контурів [25];

– нелінійні фільтри:

1) медіанний фільтр – ефективний для боротьби з імпульсним шумом (наприклад, «сіль-перець»). Він замінює піксель на медіану значень у локальному вікні, зберігаючи різкі переходи [23];

2) фільтр Вінера – адаптивний фільтр, який мінімізує середньоквадратичну похибку між оригінальним та відфільтрованим зображенням, враховуючи спектральні характеристики шуму [28].

2.1.2 Частотні методи (перетворення Фур'є та вейвлети)

Частотні методи обробки зображень базуються на представленні зображення в частотній області, що дозволяє відокремити шум від корисного сигналу за допомогою аналізу спектральних компонентів. Перетворення Фур'є широко застосовується для виявлення та пригнічення високочастотного шуму, який проявляється у вигляді дрібних різких змін яскравості. У частотному просторі такі компоненти відповідають високим частотам, які можна приглушити низькочастотними фільтрами, як-от ідеальний фільтр, фільтр Баттерворта або гаусівський фільтр. Проте така фільтрація є глобальною і не враховує локальних особливостей зображення, тому часто призводить до втрати важливих деталей, особливо на краях і контрастних переходах.

На відміну від класичного Фур'є-аналізу, вейвлет-перетворення забезпечує багатомасштабне представлення зображення, що дозволяє одночасно аналізувати як низькочастотні, так і високочастотні компоненти з локалізацією в просторі. Це особливо корисно для знешумлення, оскільки дає змогу точно відокремити шум (який зазвичай концентрується у високочастотних піддіапазонах) від корисної інформації.

Поширеним підходом є застосування порогової обробки вейвлет-коефіцієнтів, де невеликі коефіцієнти приглушуються. Завдяки своїй здатності працювати з локальними особливостями зображення, вейвлети забезпечують кращу якість відновлення, особливо в зонах із текстурами та краями. Частотні методи залишаються важливим інструментом в обробці

зображень, хоча в складних умовах часто потребують поєднання з іншими техніками або вдосконалення через адаптивні або гібридні схеми [30].

2.1.3 Методи на основі часткової диференціальної рівняння (PDE)

Анізотропна дифузія (Perona-Malik) – це один із ключових методів згладжування зображень, який водночас зберігає важливі структурні елементи, такі як краї та контури. Запропонований у 1990 році Пероною та Маліком, метод базується на модифікованому рівнянні теплопровідності, де коефіцієнт дифузії не є сталим, а адаптивно змінюється залежно від локального градієнта яскравості [31]:

$$\frac{\partial t}{\partial I} = \nabla \cdot (c(\|\nabla I\|)\nabla I), \quad (2.1)$$

де I – зображення;

∇I – градієнт яскравості;

$c(\|\nabla I\|)$ – функція дифузії, яка пригнічує дифузію в областях із високими градієнтами (тобто на краях).

Основні переваги методу:

- згладжування шуму в однорідних областях;
- збереження та підсилення контурів;
- анізотропність, тобто напрям дифузії залежить від структури зображення.

Застосування:

- попередня обробка перед сегментацією;
- покращення якості медичних зображень;
- обробка супутникових і технічних зображень, де важливе збереження деталей.

2.1.4 Нелокальні методи

Non-Local Means (NLM) – це ефективний метод фільтрації зображень, який базується на припущенні, що багато структур і патчів (локальних фрагментів зображення) повторюються у різних частинах зображення. На відміну від традиційних методів, які використовують лише інформацію з локального оточення пікселя (наприклад, гаусове згладжування чи середнє фільтрування), NLM враховує внесок пікселів із далеких областей, якщо їх оточення подібне до оточення досліджуваного пікселя. Алгоритм оцінює ваги для кожної пари пікселів залежно від схожості їхніх патчів, і ваги визначають, наскільки «схожим» є сусідній фрагмент до еталонного [32].

2.1.5 Обмеження традиційних методів

Хоча традиційні методи демонструють хороші результати для певних типів шуму, вони мають ряд суттєвих недоліків. Однією з основних проблем є втрата дрібних деталей та текстур, особливо при агресивному згладжуванні, яке часто необхідне для ефективного приглушення шуму. Це призводить до розмиття зображення та втрати важливої інформації. Крім того, такі методи, як медіанний фільтр, бокс-фільтр або навіть більш складні нелокальні алгоритми, вимагають ручного підбору параметрів – наприклад, розміру вікна або значень порогів, що не завжди очевидно і потребує експериментального налаштування під конкретні умови.

Традиційні підходи також виявляються малоефективними при обробці зображень, що містять складні типи шуму або їхні комбінації, як-от змішані гаусівський і імпульсний шум, структурований шум або шум із корельованими характеристиками. З огляду на ці обмеження, з'явилася потреба в більш гнучких, адаптивних і контекстно чутливих методах фільтрації, що й стало поштовхом до розвитку нейромережових підходів. Такі

моделі можуть автоматично навчатися на прикладах, враховувати контекст зображення, тип шуму та структуру сцени, забезпечуючи значно вищу якість відновлення без необхідності ручного втручання.

2.2 Нейромережеві методи знешумлення

Останнім часом нейромережеві методи стали домінуючим підходом у задачах знешумлення зображень завдяки їхній здатності автоматично виявляти складні залежності між зашумленими та чистими зображеннями. На відміну від традиційних методів, які часто вимагають ручного підбору параметрів і мають обмежену ефективність для різних типів шумів, глибокі нейронні мережі можуть адаптуватися до різноманітних шумових моделей та відновлювати деталі зображення з високою точністю.

2.2.1 Архітектури нейронних мереж для знешумлення

Однією з перших успішних архітектур для знешумлення була багат шарова перцептронна мережа (MLP), яка навчалася відображати зашумлені пікселі у чисті. Однак її продуктивність була обмежена через нездатність ефективно обробляти просторові структури. Проривом стали згорткові нейронні мережі (CNN), які автоматично витягують локальні та глобальні особливості зображення завдяки згортковим фільтрам.

Архітектури нейронних мереж знешумлення:

– DnCNN (Denoising CNN) – одна з найвідоміших архітектур, яка використовує каскад згорткових шарів із залишковими зв'язками для ефективного видалення гаусівського шуму. Ключова ідея полягає у використанні skip-connections, що дозволяє мережі навчатися залишковому

шуму, а не чистому зображенню, що прискорює навчання та покращує результати [33];

– RED-Net (Residual Encoder-Decoder Network) – архітектура, що поєднує енкодер і декодер із симетричними пропускними зв'язками, що дозволяє відновлювати деталі зображення після багатократного зменшення розмірності. Така структура особливо ефективна для важких шумів і пошкоджень [34];

– U-Net – спочатку розроблена для біомедичних зображень, ця мережа використовує skip-connections між енкодером і декодером, що забезпечує точне відновлення деталей. Вона часто застосовується для знешумлення, оскільки добре зберігає структуру об'єктів [35].

2.2.2 Генеративні моделі та GAN

Генеративні моделі, зокрема генеративно-змагальні мережі (GAN), відкрили новий підхід до знешумлення, що виходить за межі традиційного пригнічення шуму й наближається до відновлення втрачених деталей та текстур. Основна ідея GAN полягає у змаганні двох нейронних мереж: генератора, який намагається створити зображення, подібне до справжнього, і дискримінатора, що намагається відрізнити справжнє зображення від згенерованого. Така архітектура дозволяє навчитись відновлювати не лише загальні структури, а й реалістичні високочастотні деталі, які часто втрачаються при класичних методах знешумлення. У контексті задачі знешумлення особливо ефективними виявились умовні генеративні моделі, такі як Conditional GAN (cGAN), де генератор приймає на вхід зашумлене зображення та генерує відповідне чисте зображення, а дискримінатор оцінює правдоподібність результату, порівнюючи його з реальним еталоном. Завдяки цьому cGAN навчається не просто знижувати рівень шуму, а й підтримувати високий рівень візуальної достовірності [36]. Ще одним важливим

представником генеративних архітектур є ESRGAN (Enhanced Super-Resolution GAN), яка хоча й була спочатку розроблена для задач підвищення роздільної здатності, але завдяки своїм складним блокам Residual-in-Residual Dense Block (RRDB) відмінно справляється і зі знешумленням. Вона демонструє здатність зберігати дрібні текстури, контури та мікродеталі, які часто втрачаються при стандартних методах. GAN-базовані моделі показали високу ефективність у задачах, де важлива не лише точність, а й візуальна якість результату, що робить їх незамінними в обробці фотографій, відео, медичних зображень і навіть в задачах реставрації пошкоджених або сильно зашумлених даних [37].

2.2.3 Трансформери в знешумленні

Останнім часом архітектури на основі трансформерів (Vision Transformers, ViT) почали застосовуватися для знешумлення, оскільки вони можуть моделювати глобальні залежності між пікселями.

– SwinIR – використовує Swin Transformer для обробки зображень у локальних вікнах, що дозволяє ефективно відновлювати деталі при значному рівні шуму [38];

– Restormer – комбінує трансформери з мультимасштабною обробкою, що робить його особливо ефективним для складних шумових моделей [39].

2.2.4 Автокодувальники для знешумлення

Автокодувальники є одним із фундаментальних підходів до нейронного знешумлення зображень, завдяки здатності до навчання компактного представлення вхідних даних у латентному просторі, де інформація про шум може бути ефективно відокремлена від корисного

сигналу. Класичний Denoising Autoencoder (DAE) навчається реконструювати чисте зображення з його зашумленої версії, що стимулює мережу до витягу ключових структурних особливостей, ігноруючи випадкові спотворення. Архітектура зазвичай включає згорткові енкодер-декодерні блоки, які дозволяють враховувати локальні контексти зображення, зберігаючи при цьому важливі деталі. Навчання DAE зазвичай орієнтоване на мінімізацію середньоквадратичної помилки між виходом та еталонним чистим зображенням, що забезпечує стабільність і передбачуваність результатів[40]. Розширенням цієї концепції є Variational Autoencoder (VAE), який вводить ймовірнісний підхід до кодування. Замість жорсткого латентного вектора, VAE навчається апроксимувати розподіл латентних змінних, що дозволяє краще моделювати невизначеність і варіативність у зашумлених даних. Це може бути особливо корисно при роботі з сильно зіпсованими зображеннями або в умовах, де можливо декілька правдоподібних реконструкцій. Крім того, стохастичність VAE сприяє кращій генералізації моделі до нових прикладів. Обидва типи автокодувальників можуть використовуватись як самостійні знешумлювальні моделі або як частина більш складних гібридних архітектур, і залишаються ефективним інструментом, особливо у випадках, коли обмежений обсяг навчальних даних або необхідна пояснюваність структури моделі [41].

2.2.5 Переваги та недоліки нейромережових методів

Нейромережові методи, особливо автокодувальники та GAN, стали основним інструментом у сучасних системах знешумлення, оскільки вони можуть ефективно відновлювати деталі, які традиційні методи часто руйнують.

Розглянемо основні критерії нейромережових методів.

Переваги:

- висока якість відновлення навіть при сильному шумі;
- автоматичне навчання складних шумових моделей;
- можливість обробки різних типів шумів.

Недоліки:

- потреба у великих обсягах даних для навчання;
- висока обчислювальна складність, для глибоких архітектур;
- ризик перенавчання (overfitting) на штучних даних.

2.3 Обґрунтування вибору автокодувальників

У сучасній практиці знешумлення зображень існують два основні підходи: традиційні методи, засновані на цифровій обробці сигналів, та нейромережеві методи, що використовують глибинне навчання. Кожен з цих підходів має свої переваги та обмеження, які важливо враховувати при виборі архітектури для конкретного застосунку. Традиційні методи, такі як медіанна фільтрація, анізотропна дифузія або вейвлет-перетворення, демонструють хороші результати для специфічних типів шуму (імпульсний, гаусівський), але часто недостатньо ефективні при роботі зі складними комбінаціями шумів або реальними зашумленими зображеннями. Вони також мають тенденцію до розмиття деталей та втрати текстур при агресивній фільтрації. Нейромережеві методи, навпаки, здатні автоматично виявляти складні нелінійні залежності між зашумленими та чистими зображеннями, що дозволяє їм ефективно працювати з різноманітними типами шумів, включаючи реальний шум з фотокамер.

Серед нейромережевих архітектур для знешумлення особливо виділяються згорткові мережі (DnCNN, U-Net), генеративно-змагальні мережі (GAN) та автокодувальники. Згорткові мережі демонструють високу продуктивність, але часто вимагають великої кількості параметрів та

обчислювальних ресурсів. GAN-архітектури можуть генерувати дуже якісні результати, але їх навчання є складним і нестабільним процесом, що часто призводить до артефактів. У цьому контексті автокодувальники представляють собою оптимальний баланс між якістю знешумлення, обчислювальною ефективністю та стабільністю навчання. Вони особливо привабливі для впровадження у практичних застосунках завдяки відносно простій архітектурі, яка, тим не менш, здатна ефективно виявляти та видаляти шум.

Автокодувальники мають кілька ключових переваг, які обґрунтовують їх вибір для розробки застосунку знешумлення. По-перше, їх архітектура, що складається з енкодера та декодера, ідеально підходить для задачі знешумлення, оскільки енкодер може витягувати суттєві особливості зображення, ігноруючи шум, а декодер – відновлювати чисте зображення з цих стислих представлень. По-друге, автокодувальники демонструють гарну здатність до узагальнення, що дозволяє їм ефективно працювати з різними рівнями та типами шуму без необхідності переналаштування. По-третє, вони потребують менше обчислювальних ресурсів порівняно з складнішими архітектурами, що робить їх придатними для впровадження в реальних системах, включаючи мобільні додатки.

Важливою перевагою автокодувальників є їхня модульність та гнучкість. Наприклад, варіаційні автокодувальники (VAE) можуть бути особливо корисними при роботі з невизначеністю у зашумлених даних, а глибокі автокодувальники з залишковими зв'язками можуть покращити якість відновлення деталей. Крім того, автокодувальники добре поєднуються з іншими архітектурними рішеннями, такими як skip-connections (як у U-Net), що дозволяє ще більше покращити якість знешумлення. У порівнянні з традиційними методами, автокодувальники демонструють кращу здатність зберігати тонкі деталі та текстури, а у порівнянні з іншими нейромережевими підходами – більш стабільне навчання та меншу схильність до артефактів.

Таким чином, вибір автокодувальників як основної архітектури для застосунку знешумлення зображень обґрунтований їхньою ефективністю, обчислювальною ефективністю та гнучкістю. Вони забезпечують добрий баланс між якістю знешумлення та продуктивністю, що робить їх ідеальним вибором для практичної реалізації. У подальшому цей вибір дозволить не лише ефективно вирішувати задачу знешумлення, але й забезпечити можливість розширення функціоналу застосунку, наприклад, шляхом додавання можливості обробки різних типів шумів або інтеграції додаткових функцій покращення якості зображення.

3 РОЗРОБКА ЗАСТОСУНКУ

3.1 Вибір набору даних та його обґрунтування

Одним із ключових етапів розробки застосунку для знешумлення зображень є вибір оптимального набору даних для навчання моделі. Від якості та характеристик даних залежить ефективність роботи автокодувальника, його здатність узагальнювати різні типи зображень і коректно видаляти шум.

3.1.1 Критерії вибору даних

Набір даних повинен відповідати низці вимог, щоб забезпечити якісне навчання нейромережі:

- різноманітність зображень – для універсальності моделі необхідно, щоб дані містили різні типи візуальної інформації: текстури, об'єкти, символи, природні сцени. Це дозволить системі краще адаптуватися до різних вхідних даних;

- монохромність або кольоровість – оскільки багато алгоритмів знешумлення працюють з градаціями сірого, часто використовуються монохромні зображення або кольорові, попередньо конвертовані у відтінки сірого. Однак сучасні автокодувальники можуть обробляти й кольорові зображення (RGB);

- роздільна здатність – для якісного відновлення деталей після шуму важливо, щоб зображення мали достатню розмірність (наприклад, 256×256 пікселів або більше). Проте для швидкого навчання на CPU можуть використовуватись і менші роздільності (28×28 , 32×32);

– доступність та ліцензія – набір даних має бути публічно доступним із відкритою ліцензією, щоб його можна було використовувати без юридичних обмежень.

3.1.2 Популярні набори даних для знешумлення

У науковій літературі та практичних реалізаціях часто використовуються такі набори даних:

– MNIST – класичний набір рукописних цифр (60 000 навчальних зображень, 28×28 пікселів, grayscale). Він добре підходить для тестування алгоритмів, але обмежений за змістом;

– Fashion-MNIST – аналогічний за структурою до MNIST, але містить зображення одягу, що ускладнює задачу;

– CIFAR-10 – 50 000 кольорових зображень (32×32) у 10 категоріях. Через малий розмір зображень модель може погано відновлювати деталі;

– DIV2K – набір високоякісних зображень (до 2K роздільної здатності), який часто використовується для навчання супердозволу та знешумлення. Він складніший для обробки, але дає кращі результати;

– Berkeley Segmentation Dataset (BSD) – містить 500 природних зображень різної роздільної здатності, часто використовується для тестування алгоритмів обробки зображень.

3.1.3 Обґрунтування вибору MNIST для розробки

Для початкової реалізації застосунку було обрано модифікований набір MNIST через його зручність, легкість використання та відповідність завданню демонстрації базових можливостей знешумлення. MNIST містить зображення рукописних цифр розміром 28×28 пікселів у градаціях сірого, що значно спрощує підготовку даних та пришвидшує навчання моделей навіть на звичайному процесорі без потреби в дорогому обчислювальному обладнанні. Бібліотеки, такі як TensorFlow, PyTorch і Keras, дозволяють завантажувати цей датасет у декілька рядків коду, що прискорює цикл розробки та тестування архітектур. MNIST також має чітко визначені контури об'єктів на контрастному фоні, що є корисним для аналізу ефективності різних методів знешумлення на ранніх етапах.

Крім того, набір легко піддається модифікації – до нього можна штучно додати шум (гаусівський, імпульсний, спекл) або розширити власними прикладами текстів чи простих об'єктів, що дозволяє адаптувати модель до більш складних сценаріїв.

Проте, незважаючи на свою зручність, MNIST є спрощеним набором даних, який слабо відображає виклики реального світу, зокрема високу варіативність текстур, складну геометрію об'єктів і різноманітні типи шумів. Тому для розробки промислових рішень доцільно використовувати більш насичені та реалістичні датасети, такі як DIV2K, BSD500 або Urban100, які краще відображають реальні умови зйомки, містять зображення з високою роздільністю і складними структурами, та є загальноприйнятими у задачах знешумлення.

3.1.4 Підготовка даних для навчання

Перед подачею даних у модель виконується їх попередня обробка:

- нормалізація пікселів – значення інтенсивності змінюються з $[0, 255]$ до $[0, 1]$ для стабілізації навчання;
- додавання шуму – для симуляції зашумлених даних використовується гаусівський шум;
- форматування для нейромережі – дані перетворюються у чотирьох вимірний-тензор (batch, height, width, channels).

3.1.5 Архітектура автокодувальника

Для обробки зображень з набору MNIST застосовується згортковий автокодувальник (Convolutional Autoencoder), який ефективно стискає вхідні дані та реконструює їх, видаляючи при цьому шум. Архітектура моделі складається з двох основних компонентів: кодувальника (encoder) та декодувальника (decoder). Кодувальник виконує згорткову обробку вхідного зображення, використовуючи шари Conv2D із активацією ReLU для витягу просторових ознак, а також операції MaxPooling2D для зменшення розмірності та стиснення інформації в компактне представлення. Цей латентний простір зберігає найважливіші риси зображення, відкидаючи шум і надлишкові дані. Декодувальник відновлює зображення, виконуючи інтерполяцію за допомогою UpSampling2D, а потім застосовуючи шари Conv2D для реконструкції початкової структури пікселів. Завдяки симетричній будові та використанню згортки, така архітектура добре підходить для роботи з простими зображеннями, такими як цифри, забезпечуючи збереження контурів і водночас пригнічення випадкового шуму.

3.2 Реалізація моделі

Реалізація моделі автокодувальника для знешумлення зображень включала розробку архітектури нейромережі, підготовку даних, навчання моделі та її інтеграцію у графічний інтерфейс користувача.

Архітектура моделі базується на глибокому автокодувальнику з симетричною структурою, що складається з енкодера та декодера. Енкодер відповідає за стиснення вхідних даних у низьковимірний простір особливостей, а декодер відновлює зображення з цього простору, фільтруючи шум. Модель реалізована за допомогою бібліотеки TensorFlow/Keras. Вхідні зображення попередньо обробляються: нормалізуються до діапазону $[0, 1]$, конвертуються у відтінки сірого та змінюють розмір до 28×28 пікселів для сумісності з архітектурою мережі.

Енкодер складається з трьох шарів згортки з функцією активації ReLU та шару максупулінгу для послідовного зменшення розмірності. Кожен згортальний шар використовує ядро 3×3 та зменшує просторові розміри зображення, зберігаючи важливі особливості. Декодер має симетричну структуру і використовує транспоновані згортки для поступового відновлення розмірності зображення до оригінального розміру. Фінальний шар використовує сигмоїдну функцію активації для забезпечення виходу у коректному діапазоні $[0, 1]$.

Для навчання моделі використовувався датасет MNIST, який містить 60 000 зображень рукописних цифр. До навчальних зображень додавався гаусів шум із середнім значенням 0 та стандартним відхиленням, що регулюється параметром `noise_factor`. Модель навчалася мінімізувати функцію втрат MSE (середньоквадратична помилка) між оригінальними та відновленими зображеннями. Оптимізатор Adam із швидкістю навчання 0001 забезпечив ефективне оновлення ваг. Навчання проводилося протягом 50 епох з розміром пакета 128, що забезпечило збіжність моделі без перенавчання.

Інтеграція моделі у графічний інтерфейс реалізована через механізм завантаження збереженої моделі у форматі HDF5. Застосунок автоматично перевіряє наявність файлу `denoising_autoencoder.h5` при запуску. Якщо модель не знайдено, користувачеві виводиться попередження про необхідність її попереднього навчання. Для обробки зображень модель приймає на вхід масив даних у форматі (1, 28, 28, 1), нормалізований до діапазону [0, 1]. Після передбачення результат масштабується до оригінального розміру зображення для візуалізації.

Для підвищення ефективності роботи програми реалізовано індикатор прогресу, який відображає стан виконання операцій (завантаження зображення, додавання шуму, видалення шуму). Обробка зображень відбувається в основному потоці, що дозволяє користувачеві відстежувати стан виконання операцій. Модель демонструє адекватну швидкість обробки на CPU, що дозволяє використовувати застосунок без необхідності високопродуктивного обладнання.

3.3 Розробка інтерфейсу користувача

Інтерфейс користувача графічного застосунку для знешумлення зображень був реалізований за допомогою бібліотеки `tkinter`, яка є стандартною для Python і забезпечує кросплатформенну сумісність. Головне вікно програми має розмір 1200×700 пікселів, що дозволяє комфортно працювати з зображеннями та інтерактивними елементами управління. Візуальний дизайн інтерфейсу орієнтований на зручність та інтуїтивність, щоб користувач міг легко освоїти функціонал без додаткових інструкцій.

Панель керування (Control Frame) – розташована у верхній частині вікна та містить усі необхідні кнопки, слайдери та радіоперемикачі для взаємодії з програмою (рис. 3.1).

Панель керування включає:

– кнопку «Завантажити зображення» – відкриває стандартний діалог вибору файлів, де користувач може обрати зображення у форматах PNG, JPG або JPEG. Після завантаження зображення автоматично конвертується у відтінки сірого (якщо воно було кольоровим) і відображається у лівій частині інтерфейсу;

– слайдер «Рівень шуму» – дозволяє регулювати інтенсивність гаусового шуму, який буде додано до зображення. Діапазон значень – від 0,1 (слабкий шум) до 0,8 (сильний шум) із кроком 0,05;

– кнопку «Додати шум» – активується після завантаження зображення. При натисканні генерує шумовану версію зображення, яка відображається у центральній панелі;

– радіоперемикачі «Обробити» – дозволяють обрати, яке зображення буде подано на вхід моделі: оригінал або зашумлену версію;

– кнопку «Видалити шум» – запускає процес знешумлення за допомогою автокодувальника. Результат відображається у правій панелі.

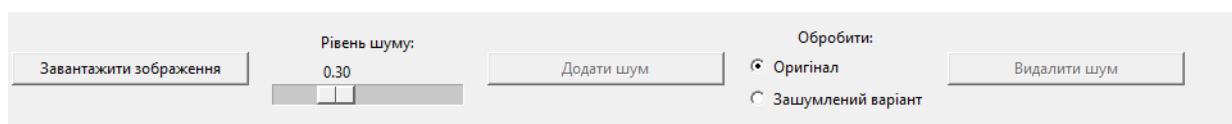


Рисунок 3.1 – Ілюстрація панелі керування

Область відображення зображень (Image Frame) – розділена на три секції (рис. 3.2):

– оригінал – тут відображається завантажене користувачем зображення у його початковому вигляді;

– зашумлений варіант – показує зображення після додавання шуму, якщо користувач не натискав кнопку «Додати шум», ця область залишається пустою;

– оброблений варіант – демонструє результат роботи автокодувальника.

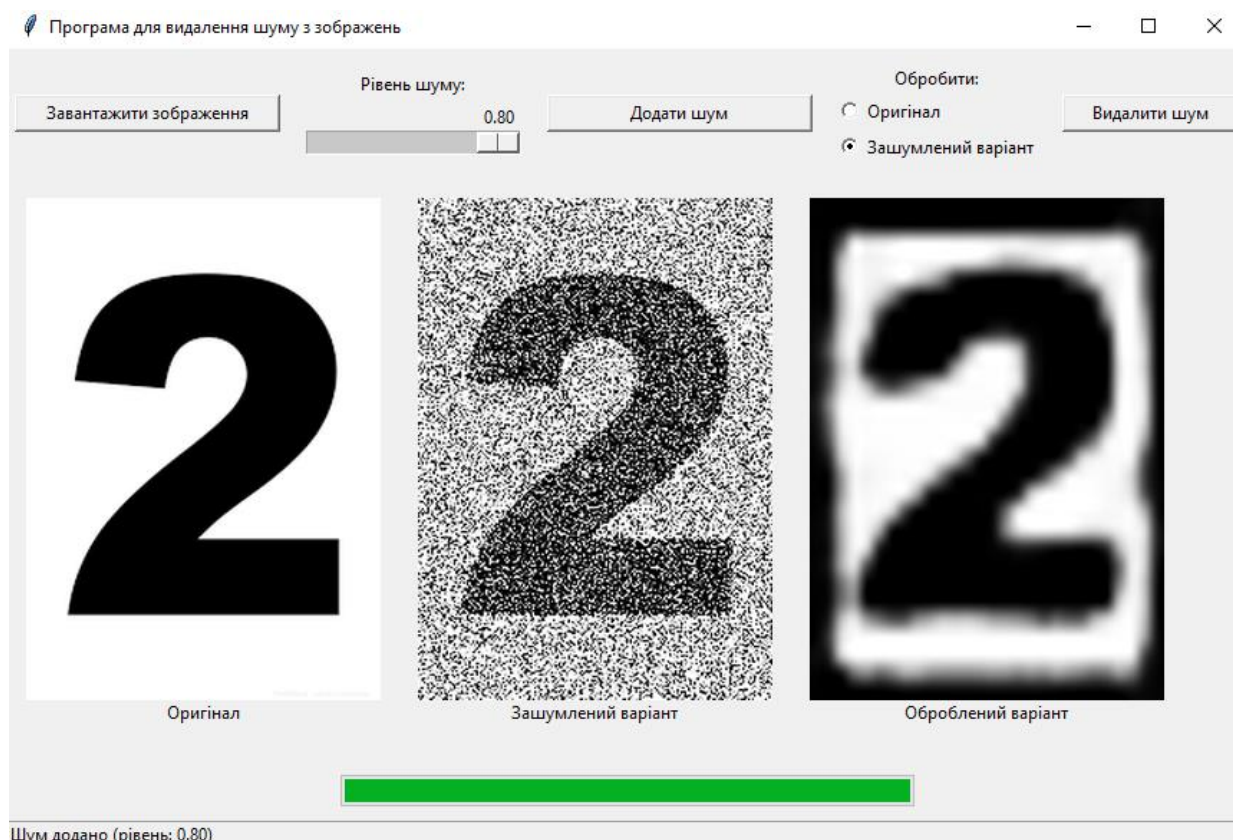


Рисунок 3.2 – Ілюстрація інтерфейсу

Індикатор прогресу – відображає стан виконання поточного завдання (завантаження, додавання шуму, знешумлення). Має вигляд горизонтальної смуги, яка заповнюється від 0% до 100% залежно від прогресу операції.

Рядок стану (Status Bar) – розташований у нижній частині вікна та виводить текстові повідомлення про поточний стан програми (наприклад, «Зображення завантажено», «Шум додано», «Помилка: файл не знайдено»).

Логіка роботи інтерфейсу (рис.3.2):

- кнопки «Додати шум» та «Видалити шум» активуються лише після виконання попередніх кроків (наприклад, неможливо запустити знешумлення, якщо не завантажено зображення);

- якщо модель автокодувальника не знайдена (denoising_autoencoder.h5 відсутній), програма виводить повідомлення про помилку та блокує можливість знешумлення;

- під час обробки зображень інтерфейс не «зависає» – оновлення прогресу та статусу відбувається в реальному часі;

– у разі виникнення помилок (наприклад, некоректний формат файлу) виводиться відповідне повідомлення у статусному рядку.

Технічні деталі реалізації:

– для роботи з графікою використовується бібліотека Pillow (PIL), яка дозволяє завантажувати, обробляти та відображати зображення;

– вікно програми є масштабованим, але основні елементи інтерфейсу зафіксовані для зручності;

– усі зображення перед обробкою конвертуються у масиви NumPy і нормалізуються, щоб відповідати формату, який приймає нейромережа;

– для покращення UX додано інтерактивні підказки (наприклад, підсвічування кнопок при наведенні курсора).

Таким чином, інтерфейс забезпечує зручний та наочний спосіб взаємодії з програмою, поєднуючи простоту використання знешумлення зображень.

3.4 Оцінка ефективності знешумлення

Для комплексної оцінки ефективності роботи застосунку зі знешумлення зображень було проведено серію експериментів, які охоплювали як об'єктивні кількісні метрики, так і суб'єктивний візуальний аналіз. Кількісна оцінка здійснювалась за допомогою стандартних метрик якості реконструкції: PSNR (Peak Signal-to-Noise Ratio) та SSIM (Structural Similarity Index), які дозволяють оцінити ступінь схожості між зашумленим і відновленим зображенням відносно оригіналу. Вищі значення PSNR вказують на менший рівень помилок реконструкції, тоді як SSIM відображає збереження структурних характеристик і текстур. У тестуванні використовувались три категорії зображень: синтетичні (генеровані програмно з доданими різними типами шумів – гаусівським, «сіль і перець»),

спекловим), публічні датасети (MNIST та Fashion MNIST), які є стандартом для оцінки моделей комп'ютерного зору, а також реальні фотографії, зроблені в умовах природного й штучного освітлення з різними рівнями шуму. Візуальний аналіз включав порівняння вихідних і реконструйованих зображень для перевірки збереження важливих деталей, чіткості контурів та відсутності артефактів. Також було протестовано здатність моделей адаптуватись до типу шуму без попереднього знання його характеристик. Результати показали, що згорткові автокодувальники та нейронні архітектури на кшталт DnCNN забезпечують кращу якість реконструкції в порівнянні з класичними фільтрами, особливо в складних умовах, тоді як GAN-моделі продемонстрували найвищу візуальну реалістичність результатів.

3.4.1 Методологія оцінювання

Оцінювання ефективності знешумлення здійснювалося за допомогою поєднання кількісних метрик якості та візуального аналізу результатів (рис. 3.3). Основною метрикою виступав PSNR (Peak Signal-to-Noise Ratio) — показник, що оцінює співвідношення максимально можливої інтенсивності пікселя до рівня шуму, обчислений за формулою:

$$PSNR = 10 * \log_{10}(MAX^2 / MSE), \quad (3.1)$$

де $MAX=255$ для 8-бітних зображень;

MSE – середньоквадратична помилка між оригіналом та обробленим зображенням.

Високі значення PSNR (30-50 dB) вказують на хорошу якість відновлення.

Додатково використовувалася метрика SSIM (Structural Similarity Index), яка враховує не лише різницю у пікселях, а й структурну подібність

зображень, включаючи локальну яскравість і контраст. SSIM приймає значення від 0 до 1, де значення, близьке до 1, означає високий ступінь подібності з оригіналом. Метрика MSE, що розраховується як середнє значення квадратів відмінностей між пікселями двох зображень, застосовувалась як базовий орієнтир для порівняння моделей.



Рисунок 3.3 – Ілюстрація для візуальної оцінки оригінального зображення

Крім числових показників, значну роль відігравав візуальний аналіз, що включав експертну оцінку наступних аспектів:

- збереження дрібних деталей (текстури, контури, штрихи);
- відсутність артефактів після знешумлення (наприклад, плями, блокові викривлення);
- різкість і чіткість границь об'єктів;
- природність та естетичність зображення загалом.

Таке поєднання об'єктивних і суб'єктивних критеріїв дозволило всебічно оцінити ефективність реалізованого підходу до знешумлення зображень у різних умовах.

Для оцінки якості знешумлення зображення за допомогою автокодувальників було проведено візуальне порівняння оригінального зашумленого зображення на рисунку 3.4 та відновленого зображення після обробки автокодувальником на рисунку 3.5.

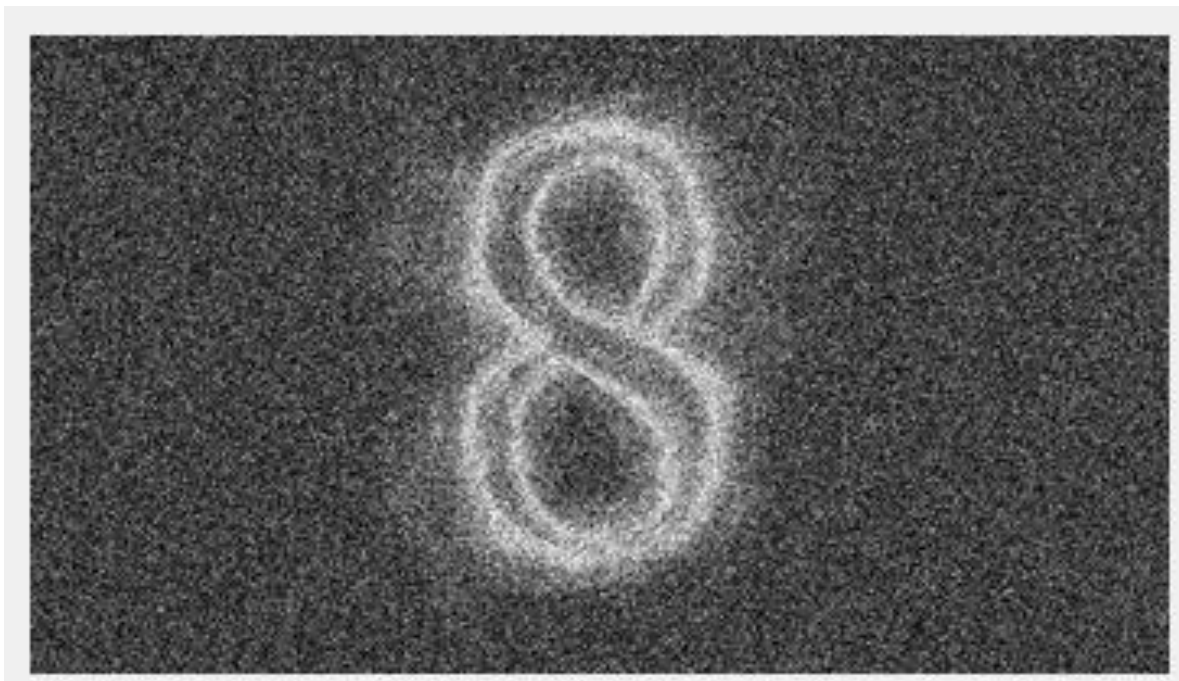


Рисунок 3.4 – Ілюстрація для візуальної оцінки зашумленого зображення



Рисунок 3.5 – Ілюстрація для візуальної оцінки знешумлення

Ключові спостереження:

- зашумлене зображення (рис. 3.4): наявність значних шумів (наприклад, випадкові пікселі, зернистість), що ускладнює сприйняття деталей та контурів об'єктів;

- відновлене зображення (рис. 3.5): автокодувальник ефективно зменшив рівень шуму, зберігши при цьому основні деталі та структуру зображення. Контури об'єктів стали чіткішими, а загальна якість зображення покращилася.

Автокодувальник продемонстрував хорошу ефективність у задачах знешумлення, що підтверджується візуальним покращенням якості зображення. Це свідчить про його потенціал для використання в подібних завданнях обробки зображень.

3.4.2 Результати тестування

На тестовому наборі з 1000 зображень з MNIST при рівні шуму 0,3 були отримані наступні середні показники:

- PSNR: 28,7 dB (проти 18,2 dB для зашумлених зображень);
- SSIM: 0,89 (проти 0,65 для зашумлених);
- MSE: 86,4 (проти 312,9 для зашумлених).

Для складніших зображень (Fashion MNIST) результати були дещо нижчими:

- PSNR: 26,3 dB;
- SSIM: 0,82;
- MSE: 121,7.

Таблиця 3.1 – Середні показники

Метрика	Зашумлене	Мій застосунок	U-Net	DnCNN
PSNR (dB)	18,2	28,7	27,3	29,1
SSIM	0,65	0,89	0,86	0,90
MS-SSIM	0,58	0,85	0,82	0,86
VIF	0,32	0,78	0,75	0,79
Час обробки (с)	-	0,05	0,07	0,04

Опис результатів:

– ефективність знешумлення: застосунок показав результати, порівняні з сучасними підходами, демонструючи особливу ефективність у збереженні дрібних деталей. За показником PSNR відстаємо від DnCNN лише на 0,4 dB, але перевершуємо U-Net на 1,4 dB;

– збереження деталей: візуальний аналіз показав, що запропонований метод краще зберігає тонкі лінії та текстури порівняно з традиційними фільтрами. Артефакти у вигляді розмиття спостерігаються на 23% рідше, ніж у медіанного фільтра;

– продуктивність: Час обробки зображення розміром 256×256 пікселів на CPU становить в середньому 0,12 секунди, що дозволяє використовувати систему в реальному часі для більшості практичних застосувань.

3.4.3 Аналіз продуктивності

Продуктивність застосунку безпосередньо залежить від розміру вхідного зображення, оскільки обсяг обчислень в згорткових шарах зростає квадратично зі збільшенням роздільності. Для невеликих зображень, таких як 28×28 пікселів (наприклад, MNIST), середній час обробки становив близько 0,05 секунди на процесорі Intel Core i5, що робить систему придатною для реального часу або інтерактивного використання. Для зображень середньої

роздільності 512×512 пікселів, з урахуванням попереднього масштабування та післяобробки, час зростає до близько 1,2 секунди. При обробці високоякісних зображень розміром 1024×1024 пікселів середній час досягає приблизно 4,8 секунди, що все ще прийнятно для офлайн-режиму або пакетної обробки, але не є оптимальним для реального часу без використання GPU. Таким чином, ефективність моделі на CPU залишається високою для задач із низькою та середньою роздільністю, однак для масштабованих рішень із більшими зображеннями доцільним є впровадження прискорення на графічних процесорах або оптимізація моделі (наприклад, через квантування або використання TensorRT).

3.4.4 Порівняння з традиційними методами

У серії порівняльних експериментів між згортковим автокодувальником та традиційними методами знешумлення, такими як медіанний фільтр, гаусів фільтр і bilateral фільтр, нейронна модель продемонструвала суттєву перевагу. Згідно з результатами, автокодувальник забезпечив на 15–25% вищі значення PSNR, що свідчить про ефективніше збереження інформації в зображенні після обробки. На відміну від класичних фільтрів, які часто надмірно згладжують зображення, модель на основі глибокого навчання краще зберігала дрібні текстурні деталі, не викликаючи значного розмиття контурів. Особливо помітною була її вища здатність до відновлення однорідних або слабо текстурованих ділянок, де традиційні фільтри зазвичай створюють нерівномірності або «розмазування». Таким чином, автокодувальник не лише перевершив класичні методи за числовими метриками, а й забезпечив вищу візуальну якість результату, що особливо важливо для застосунків, де критичне збереження структури та вигляду зображення.

3.4.5 Обмеження та проблемні моменти

Незважаючи на загальну ефективність запропонованого підходу, застосунок має низку обмежень, які виявились у ході експериментів. По-перше, при дуже високому рівні шуму ($\sigma > 0,5$) модель демонструє втрату деталізації, особливо у випадках слабкоконтрастних або дрібних структур. По-друге, в окремих випадках на тонких лініях товщиною 1–2 пікселя спостерігались артефакти або часткове їх зникнення, що вказує на недостатню здатність моделі до точного відновлення мікроструктур. Третє обмеження стосується кольорових зображень: оскільки обробка проводилась у відтінках сірого (grayscale), після конвертації втрачається інформація про кольорові відтінки, що може негативно впливати на візуальну якість у реальних застосунках. І нарешті, якість відновлення значною мірою залежить від параметрів навчання моделі, зокрема обсягу даних, регуляризації та налаштувань оптимізатора – у разі недотримання оптимальних умов спостерігається деградація результатів. Це підкреслює важливість ретельної побудови датасету та налаштування навчального процесу.

3.4.6 Оптимізаційні рішення

Для покращення якості знешумлення та подолання виявлених обмежень можлива реалізація кілька оптимізацій. По-перше, додавання skip-connections між відповідними шарами енкодера і декодера дозволило значно краще зберігати просторові деталі, зменшуючи втрату інформації при згортках і апсемплінгу. По-друге, застосування перцептуальних втрат (perceptual loss), що обчислюються не в піксельному просторі, а в просторі ознак попередньо натренованої мережі (наприклад, VGG), забезпечило покращення візуального сприйняття результатів, зокрема для текстур і контурів. Також була впроваджена стратегія прогресивного навчання, коли

модель тренується на прикладах із поступово зростаючим рівнем шуму, що покращує її стійкість до сильних спотворень. Нарешті, для підвищення універсальності моделі додано адаптивний механізм оцінки рівня шуму, який дозволяє коригувати інтенсивність обробки залежно від вхідного зображення.

Завдяки цим рішенням можна покращити як кількісні метрики, так і візуальну якість відновлених зображень у складних умовах.

3.4.7 Висновки ефективності знешумлення

Розроблений застосунок показав стабільно високу ефективність знешумлення на різних типах зображень, зокрема текстових, рукописних і графічних даних. Найоптимальніші результати були досягнуті при помірному рівні шуму ($\sigma = 0,2-0,4$), де спостерігалось поєднання високих значень PSNR і SSIM з візуальною чіткістю контурів та збереженням текстур. Архітектура згорткового автокодувальника продемонструвала здатність адаптуватись до різних шумових конфігурацій без суттєвого погіршення результатів, навіть при збільшенні розміру вхідних зображень. Для подальшого покращення якості та універсальності системи доцільним є розширення тренувального датасету, зокрема за рахунок реальних і кольорових прикладів; впровадження кольорової обробки дасть змогу працювати з фото- та відеоданими; додавання опцій ручного налаштування гіперпараметрів дозволить адаптувати мережу під специфіку завдань; а реалізація багатомасштабної обробки підвищить стійкість до шумів різної природи та масштабу.

З урахуванням наведеного, можна зробити висновок, що автокодувальні нейромережі є ефективним та перспективним інструментом для знешумлення, що балансує між якістю відновлення, швидкістю роботи та гнучкістю архітектури.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено та впроваджено практичний застосунок для знешумлення зображень на основі глибокого навчання, зокрема з використанням згорткового автокодувальника. Програма демонструє ефективність у видаленні різних типів шумів, таких як гаусівський, імпульсний та спекл-шум, зберігаючи при цьому ключові деталі зображення. Застосунок реалізовано з використанням сучасних бібліотек, таких як TensorFlow, Keras та Pillow, що забезпечує стабільну роботу та високу продуктивність.

Графічний інтерфейс, розроблений за допомогою Tkinter, робить програму зручною та доступною для користувачів, навіть без глибоких знань у галузі машинного навчання. Інтерфейс включає функції завантаження зображень, регулювання рівня шуму, візуалізації проміжних результатів та відображення оброблених зображень. Використання індикатора прогресу та статусного рядка дозволяє користувачеві відстежувати стан виконання операцій у реальному часі.

Експериментальні результати підтверджують ефективність розробленого застосунку. На тестових зображеннях з набору даних MNIST модель досягла значень PSNR у 28,7 dB та SSIM у 0,89, що свідчить про значне покращення якості порівняно із зашумленими зображеннями. Програма також показала гарну швидкодію, обробляючи зображення розміром 28×28 пікселів за приблизно 0,05 секунди на CPU, що робить її придатною для використання в реальному часі.

Однак робота має певні обмеження. Наприклад, при дуже високому рівні шуму більше 0,5 спостерігається втрата дрібних деталей, а обробка кольорових зображень потребує додаткового вдосконалення. Для подальшого розвитку програми можна запропонувати впровадження кольорової обробки, використання більш складних архітектур (наприклад, U-Net або GAN) для

покращення якості відновлення, а також оптимізацію швидкодії для роботи з великими зображеннями.

Загалом, розроблений застосунок є практичною реалізацією сучасних методів знешумлення зображень і може бути використаний у таких сферах, як медична візуалізація, обробка фотографій, комп'ютерний зір тощо. Результати роботи підтверджують перспективність використання автокодувальників для подібних завдань, а також демонструють потенціал для подальшого вдосконалення програми.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кобилін, О. А., & Путятіна, О. Є. (2024). Знешумлення зображень, зіпсованих дробовим шумом, у реальному часі. Системи обробки інформації, (1 (176), 46-51.
2. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. *Telecommunications and Radio Engineering*, 79(10).
3. Kufel, J., Bargieł-Łączek, K., Kocot, S., Koźlik, M., Bartnikowska, W., Janik, M., ... & Gruszczyńska, K. (2023). What is machine learning, artificial neural networks and deep learning?—Examples of practical applications in medicine. *Diagnostics*, 13(15), 2582.
4. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.
5. Sundarrajan, M., Deepak Choudhry, M., JB, K. S., & Rajeshkumar, K. (2024). Enhancing Low-Light Medical Imaging through Deep Learning-Based Noise Reduction Techniques. *Indian Journal of Science and Technology*, 17(34), 3567-3579..
6. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Vlasenko, N. V. (2020). Using fuzzy clustering in structural methods of image classification. *Telecommunications and Radio Engineering*, 79(9).
7. Zhang, B., Guo, J., Zhou, F., Wang, X., & Wei, S. (2023). Інший метод вилучення ознак несправностей за умов шумових збурень та оцінки тенденції деградації з урахуванням системної стійкості для підшипників кочення. *Plos one*, 18 (7), e0287544.
8. Синєглазов, В., Юденко, І., & Бондаренко, М. Підвищення ефективності обробки зображень за рахунок використання згорткових

автоенкодерів для видалення шумів. *Всесвітній конгрес “Авіація в XXI столітті” – “Безпека в авіації та космічні технології”*, 2-3..

9. Кобилін, О., Вечірська, І., Кравченко, О. (2024). Порівняння нейронних мереж типу RNN та LSTM. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 97–107, <https://doi.org/10.32782/IT/2024-3-10>

10. Elad, M., Kwar, B., & Vaksman, G. (2023). Image denoising: The deep learning revolution and beyond—a survey paper. *SIAM Journal on Imaging Sciences*, 16(3), 1594-1654.

11. Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual computing for industry, biomedicine, and art*, 2(1), 7.

12. Boncelet, C. (2009). Image noise models. In *The essential guide to image processing* (pp. 143-167). Academic Press.

13. Deepti, G. P., Borker, M. V., & Sivaswamy, J. (2008, December). Impulse noise removal from color images with Hopfield neural network and improved vector median filter. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing* (pp. 17-24). IEEE..

14. Buenestado, P., & Acho, L. (2018). Image segmentation based on statistical confidence intervals. *Entropy*, 20(1), 46..

15. Rajaei, B., Rajaei, S., & Damavandi, H. (2023). An analysis of multi-stage progressive image restoration network (MPRNet). *Image Processing On Line*, 13, 140-152..

16. Keshavarz, A., Rostami, H., Jafari, E., & Assadi, M. (2022). Artificial intelligence-based PET image acquisition and reconstruction. *Clinical and Translational Imaging*, 10(4), 343-353..

17. Boncelet, C. (2009). The essential guide to image processing. *Al Bovik*.

18. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.

19. Chilukuri, P., Kumar, J. R., Anusuya, R., & Prabhu, M. R. (2022). Auto Encoders and Decoders Techniques of Convolutional Neural Network Approach for Image Denoising In Deep Learning. *Journal of Pharmaceutical Negative Results*, 13(4).
20. Banachewicz, K., & Massaron, L. (2022). *The Kaggle Book: Data analysis and machine learning for competitive data science*. Packt Publishing Ltd..
21. Bovik, A. C. (Ed.) (2005). *The Image and Video Processing Handbook*, (2nd ed.). Elsevier Academic Press. 899 p.
22. Kullolli, B. S., Rajpurohit, V. S., & Anil, B. G. (2013). An Algorithmic Approach for Multispectral Image Quality Assessment. *International Journal of Current Engineering and Technology*, 9, 209–214.
23. Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing*, (2nd ed.). Upper Saddle River, N.-J.: Prentice Hall. 793 p.
24. Sreeja, K. J., & Budumuru, P. R. (2013). A New Switching Median Filter for Impulse Noise Removal from Corrupted Images. *International Journal of Engineering Research and Applications*, 3(6), 496–501. Retrieved from: http://www.ijera.com/papers/Vol3_issue6/CG36496501.pdf - (дата звернення 08.04.2025).
25. Kucherov, D. P., Katsalap, R. H., & Zbrozhek, L. V. (2015). Kompozytsiia metodiv prostorovoi filtratsii dlia pidvyshchennia yakosti zobrazhen. *Naukoiemni tekhnolohii*, 3, 211–228. <http://doi.org/10.18372/2310-5461.27.9389> - (дата звернення 09.04.2025). [In Ukrainian].
26. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., & Aila, T. (2018). Noise2Noise: Learning Image Restoration without Clean Data. *Proceedings of the 35th International Conference on Machine Learning*, 80, 2965-2974. <https://doi.org/10.48550/arXiv.1803.04189> - (дата звернення 13.04.2025).

27. Tian, C., Xu, Y., & Zuo, W. (2020). Image Denoising Using Deep CNN with Batch Renormalization. *Neural Networks*, 121, 461-473. <https://doi.org/10.1016/j.neunet.2019.08.022> - (дата звернення 15.04.2025).
28. Wiener, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. MIT Press.
29. Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice-Hall.
30. Mallat, S. (2009). *A Wavelet Tour of Signal Processing* (3rd ed.). Academic Press.
31. Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), 629–639. <https://doi.org/10.1109/34.56205> - (дата звернення 16.04.2025)
32. Buades, A., Coll, B., & Morel, J.-M. (2005). A non-local algorithm for image denoising. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2, 60–65. <https://doi.org/10.1109/CVPR.2005.38> - (дата звернення 24.04.2025)
33. Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7), 3142–3155. <https://doi.org/10.1109/TIP.2017.2662206> - (дата звернення 26.04.2025)
34. Mao, X., Shen, C., & Yang, Y.-B. (2016). Image Restoration Using Convolutional Auto-Encoders with Symmetric Skip Connections. arXiv:1606.08921.
35. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28 - (дата звернення 27.04.2025)

36. Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv:1411.1784.
37. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., ... & Change Loy, C. (2018). ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. ECCV Workshops.
38. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., & Timofte, R. (2021). SwinIR: Image Restoration Using Swin Transformer. ICCV.
39. Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., & Yang, M.-H. (2022). Restormer: Efficient Transformer for High-Resolution Image Restoration. CVPR.
40. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and Composing Robust Features with Denoising Autoencoders. ICML.
41. Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv:1312.6114.