

**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Оптимізація процесу відновлення даних із шумом  
шляхом використання архітектури варіаційного  
автокодувальника»

на здобуття освітнього ступеня магістра  
зі спеціальності 121 Інженерія програмного забезпечення  
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання  
ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ Владислав ТРОЯН  
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-61  
Владислав ТРОЯН

Керівник: Вікторія КОРЕЦЬКА  
\_\_\_\_\_  
канд. пед. наук., доц.

Рецензент: \_\_\_\_\_  
науковий ступінь, Ім'я, ПРІЗВИЩЕ  
вчене звання

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

«\_\_\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Трояну Владиславу Миколайовичу

1. Тема кваліфікаційної роботи: «Оптимізація процесу відновлення даних із шумом шляхом використання архітектури варіаційного автокодувальника»

керівник кваліфікаційної роботи Вікторія КОРЕЦЬКА, канд. пед. наук., доц.,

затвердені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи «19» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, набір цифрових зображень; параметри нейронної мережі та алгоритмів оптимізації; метод вилучення та порівняння ознак; вимоги до точності класифікації та швидкодії системи.

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

1. Аналіз існуючих методів відновлення даних із шумом.

2. Методологія оптимізації процесу відновлення даних на основі VAE.

3. Розробка та валідація оптимізованого процесу відновлення даних із шумом шляхом використання архітектури варіаційного автокодувальника.

5. Перелік ілюстративного матеріалу: *презентація*

1. Математична модель оптимізованого VAE
2. Архітектура моделі.
3. Алгоритм оптимізації.
4. Практичний результат.
5. Практичний результат.
6. Порівняльна ефективність.
7. Висновки.
8. Публікації та апробації роботи.

6. Дата видачі завдання «31» жовтня 2025 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи   | Строк виконання етапів роботи | Примітка |
|-------|---|-------------------------------|----------|
| 1     | Аналіз наявної науково-технічної літератури   | 31.10-10.11.2025              |          |
| 2     | Вивчення матеріалів для аналізу методів відновлення шуму.   | 15.11-18.11.2025              |          |
| 3     | Дослідження методів відновлення шуму.   | 19.11-22.11.2025              |          |
| 4     | Аналіз особливостей впливу методів оптимізації на якість відновлення шуму.                                    | 23.11-24.11.2025              |          |
| 5     | Дослідження технологій оптимізації за допомогою архітектури варіаційного автокодувальника.                    | 18.11-25.11.2025              |          |
| 6     | Застосування методів відновлення даних із шумом шляхом використання архітектури варіаційного автокодувальника | 26.11-10.12.2025              |          |
| 7     | Оформлення роботи: вступ, висновки, реферат   | 13.12-17.12.2025              |          |
| 8     | Розробка демонстраційних матеріалів   | 18.12-19.12.2025              |          |

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Владислав ТРОЯН

Керівник

кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Вікторія КОРЕЦЬКА





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 82 стор., 2 табл., 10 рис., 30 джерело.

*Мета дослідження*- полягає в розробленні та обґрунтуванні підходу до підвищення якості відновлення зашумлених даних шляхом застосування та оптимізації архітектури варіаційного автокодувальника (VAE), а також у визначенні параметрів моделі, які забезпечують найкращі результати реконструкції.

*Об'єкт дослідження* - процес відновлення та реконструкції даних, спотворених шумом, у системах машинного навчання.

*Предмет дослідження* - методи та архітектурні особливості варіаційних автокодувальників, а також їх оптимізація з метою підвищення точності та стійкості відновлення зашумлених даних.

У роботі проведено комплексний аналіз проблематики відновлення зашумлених даних та обґрунтовано переваги варіаційних автокодувальників порівняно з традиційними методами фільтрації. Досліджено сучасні архітектурні рішення VAE, проаналізовано методи регуляризації латентного простору та оптимізації процесу навчання. Виявлено ключові проблеми базових реалізацій VAE, зокрема "зникнення" KL-дивергенції та обмежену ефективність при високих рівнях шуму.

Розроблено модифіковану архітектуру VAE, що включає розділення латентного простору на семантичні та шумові компоненти, впровадження механізмів уваги у декодер та використання skip-з'єднань між енкодером і декодером. Запропоновано композитну функцію втрат з адаптивними коефіцієнтами, що поєднує компоненти реконструкції, регуляризації, моделювання шуму та збереження структурних характеристик.

Реалізовано програмну систему на основі фреймворків TensorFlow та PyTorch, що включає модулі попередньої обробки даних, навчання моделі, візуалізації результатів та оцінки якості. Проведено експериментальні дослідження на

стандартних наборах даних з різними типами шуму (гаусівський, імпульсний, спекл). Результати демонструють підвищення якості відновлення на 15-25% за метриками PSNR та SSIM порівняно з базовими реалізаціями VAE та традиційними методами фільтрації.

Розроблено рекомендації щодо налаштування параметрів моделі для різних типів даних та рівнів шуму. Визначено перспективи подальшого розвитку дослідження, зокрема можливості інтеграції з дифузійними моделями, розширення для обробки відеопотоків та застосування в реальних системах комп'ютерного зору.

Результати дослідження мають практичну цінність для розробників систем обробки зображень, медичної візуалізації, обробки аудіосигналів та наукових досліджень. Впровадження запропонованих рішень дозволить підвищити якість відновлення даних у умовах реального шуму, забезпечити адаптивність алгоритмів до різних типів спотворень та покращити ефективність подальшого аналізу інформації.

**КЛЮЧОВІ СЛОВА:** VAE, ВАРІАЦІЙНИЙ АВТОКОДУВАЛЬНИК, ОПТИМІЗАЦІЯ АРХІТЕКТУРИ, ВІДНОВЛЕННЯ ДАНИХ, ГЛИБОКЕ НАВЧАННЯ, КОМПОЗИТНА ФУНКЦІЯ ВТРАТ, ЛАТЕНТНИЙ ПРОСТІР, ADAPTIVE LEARNING, PSNR, SSIM.

## ABSTRACT

Text part of the master's qualification work: 82 pages, 10 pictures, 2 table, 30 sources.

The purpose of the work- consists in developing and substantiating an approach to improving the quality of noisy data recovery by applying and optimizing the architecture of a variational autoencoder (VAE), as well as determining the model parameters that provide the best reconstruction results.

Object of research- the process of restoring and reconstructing noise-distorted data in machine learning systems

Subject of research- methods and architectural features of variational autoencoders, as well as their optimization to improve the accuracy and robustness of noisy data restoration.

The paper provides a comprehensive analysis of the problem of noisy data recovery and substantiates the advantages of variational autoencoders over traditional filtering methods. Modern VAE architectural solutions are investigated, and methods for regularizing latent space and optimizing the training process are analyzed. Key problems of basic VAE implementations are identified, in particular, the “disappearance” of KL divergence and limited effectiveness at high noise levels.

A modified VAE architecture has been developed, which includes the separation of the latent space into semantic and noise components, the introduction of attention mechanisms in the decoder, and the use of skip connections between the encoder and decoder. A composite loss function with adaptive coefficients is proposed, combining components of reconstruction, regularization, noise modeling, and preservation of structural characteristics.

A software system based on the TensorFlow and PyTorch frameworks has been implemented, which includes modules for preprocessing data, training the model, visualizing results, and evaluating quality. Experimental studies were conducted on standard datasets with different types of noise (Gaussian, impulse, speckle). The results

demonstrate a 15-25% improvement in restoration quality according to PSNR and SSIM metrics compared to basic VAE implementations and traditional filtering methods.

Recommendations have been developed for configuring model parameters for different types of data and noise levels. Prospects for further research development have been identified, including the possibility of integration with diffusion models, expansion for video stream processing, and application in real-world computer vision systems.

The research results are of practical value for developers of image processing systems, medical imaging, audio signal processing, and scientific research. The implementation of the proposed solutions will improve the quality of data recovery in real noise conditions, ensure the adaptability of algorithms to different types of distortions, and improve the efficiency of further information analysis.

**KEYWORDS:** VAE, VARIATIONAL AUTOCODER, ARCHITECTURE OPTIMIZATION, DATA RECOVERY, DEEP LEARNING, COMPOSITE LOSS FUNCTION, LATENT SPACE, ADAPTIVE LEARNING, PSNR, SSIM.

## Зміст

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....  | 22 |
| ВСТУП.....  | 23 |
| 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ВІДНОВЛЕННЯ ДАНИХ ІЗ ШУМОМ.....                             | 15 |
| 1.1. Поняття шуму та його вплив на якість даних. ....                                 | 15 |
| 1.2. Методи фільтрації та попереднього очищення даних. ....                           | 18 |
| 1.3. Традиційні та сучасні підходи до реконструкції сигналів і зображень. ....        | 24 |
| 1.4 Автокодувальники .....  | 28 |
| 1.5 Варіаційні автокодувальники .....   | 31 |
| 2 МЕТОДОЛОГІЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ВІДНОВЛЕННЯ ДАНИХ НА ОСНОВІ VAE .....               | 36 |
| 2.1. Постановка задачі та формальний опис моделі. ....                                | 36 |
| 2.2. Архітектура варіаційного автокодувальника для роботи із зашумленими даними. .... | 37 |
| 2.3. Методи оптимізації.....  | 40 |
| 2.4 Вибір інструментів, середовища програмування та бібліотек. ....                   | 45 |
| 2.5. Формування наборів даних та методи їх штучного зашумлення. ....                  | 48 |
| 3 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА .....  | 52 |
| 3.1. Опис процесу навчання моделі.....  | 52 |
| 3.2. Порівняння різних конфігурацій архітектури. ....                                 | 55 |
| 3.3. Аналіз впливу параметрів VAE на якість реконструкції. ....                       | 62 |
| 3.4. Порівняння VAE з іншими методами відновлення даних. ....                         | 65 |
| 3.5. Оцінювання ефективності моделі за метриками.....                                 | 66 |
| 3.6. Інтерпретація отриманих результатів. ....  | 69 |
| ВИСНОВКИ .....  | 71 |
| ПЕРЕЛІК ПОСИЛАНЬ.....   | 73 |
| ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....   | 75 |
| ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ МОДУЛІВ.....  | 81 |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AE - Autoencoder (автокодувальник).

DAE - Denoising Autoencoder (автокодувальник, навчений відновлювати чисті дані із зашумлених вхідних зразків).

VAE - Variational Autoencoder (варіаційний автокодувальник з регуляризованим латентним простором).

CNN - Convolutional Neural Network (згорткова нейронна мережа).

MSE - Mean Squared Error (середньоквадратична похибка).

MAE - Mean Absolute Error (середня абсолютна похибка).

PSNR - Peak Signal-to-Noise Ratio (пікове співвідношення сигнал/шум).

SSIM - Structural Similarity Index (індекс структурної подібності).

MS-SSIM - (багатомасштабний індекс структурної подібності).

LR - Learning Rate (швидкість навчання).

GPU - Graphics Processing Unit (графічний процесор).

ELBO - Evidence Lower Bound (нижня межа логарифмічної правдоподібності).

## ВСТУП

У сучасному цифровому середовищі проблема відновлення даних із шумом набуває все більшої актуальності через стрімке зростання обсягів інформації та різноманіття джерел її отримання. Від електронних листів до медичних зображень, від аудіозаписів до наукових вимірювань – шум є неминучим супутником будь-яких даних, що передаються або зберігаються. Традиційні методи фільтрації та відновлення часто виявляються недостатньо ефективними при роботі зі складними нелінійними залежностями та високорівневими спотвореннями.

Останнім часом глибоке навчання відкрило нові горизонти в обробці даних, пропонуючи потужні інструменти для вирішення складних завдань відновлення інформації. Серед цих інструментів особливе місце займають варіаційні автокодувальники (VAE) – генеративні моделі, що поєднують у собі переваги нейронних мереж та байєсівських методів. Відмінність VAE полягає в їх здатності не лише відновлювати дані, але й моделювати їхній глибинний розподіл, що робить їх особливо ефективними для роботи з неповними або сильно зашумленими вхідними сигналами.

Актуальність дослідження оптимізації процесів відновлення даних за допомогою VAE зумовлена потребою у створенні надійних систем, здатних ефективно працювати в умовах реального світу, де якість вхідних даних часто далека від ідеалу. Медична діагностика, автономні системи, обробка природної мови – це лише декілька прикладів галузей, де якісне відновлення інформації може мати вирішальне значення.

Основною метою дослідження є розробка оптимізованої архітектури VAE, здатної ефективно відновлювати дані зі значною часткою шуму, зберігаючи при цьому ключові характеристики оригінальної інформації. Проаналізуємо різні підходи до побудови енкодерів та декодерів, дослідимо вплив розмірності latent-простору на якість реконструкції та запропонуємо методи комбінування VAE з іншими підходами для досягнення максимальної ефективності.

Мета дослідження- полягає в розробленні та обґрунтуванні підходу до підвищення якості відновлення зашумлених даних шляхом застосування та оптимізації архітектури варіаційного автокодувальника (VAE), а також у визначенні параметрів моделі, які забезпечують найкращі результати реконструкції.

Об'єкт дослідження -процес відновлення та реконструкції даних, спотворених шумом, у системах машинного навчання.

Предмет дослідження -методи та архітектурні особливості варіаційних автокодувальників, а також їх оптимізація з метою підвищення точності та стійкості відновлення зашумлених даних. Для досягнення мети вирішувалися наступні завдання:

Для досягнення поставленої мети в роботі було визначено та вирішено такі завдання:

Виконати аналітичний огляд сучасних наукових публікацій та існуючих підходів до відновлення зашумлених даних, зокрема класичних методів фільтрації, нейромережових моделей та генеративних архітектур, з метою визначення їхніх переваг та обмежень.

Провести експериментальне дослідження ефективності різних методів відновлення зображень із шумом на репрезентативних наборах даних із використанням кількісних метрик якості.

Розробити та реалізувати оптимізовану архітектуру варіаційного автокодувальника для задачі денойзингу зображень, орієнтовану на підвищення точності реконструкції та збереження візуальних деталей.

Провести кількісний аналіз ефективності запропонованого методу з використанням метрик MSE, PSNR та SSIM, а також виконати порівняння з альтернативними підходами.

Виконати порівняльну характеристику різних конфігурацій архітектури варіаційного автокодувальника та визначити найбільш ефективні параметри моделі.

На основі отриманих експериментальних результатів удосконалити обраний підхід до відновлення зашумлених зображень з метою підвищення стійкості моделі до різних типів шуму та покращення узагальнювальної здатності.

Реалізація зазначених завдань створила надійну теоретико-експериментальну основу для дослідження та подальшого розвитку сучасних методів відновлення зашумлених цифрових зображень із використанням архітектури варіаційного автокодувальника.

# 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ВІДНОВЛЕННЯ ДАНИХ ІЗ ШУМОМ

## 1.1. Поняття шуму та його вплив на якість даних.

У процесі формування, передавання та зберігання цифрових даних завжди присутні спотворення, які виникають під впливом різноманітних фізичних, технічних або інформаційних факторів. Ці випадкові або систематичні відхилення від істинного сигналу називають шумом. Шум є невід'ємною складовою будь-якої реальної системи обробки інформації[1,2], а його присутність істотно впливає на точність, інформативність і достовірність отриманих даних.

Шум може виникати на будь-якому етапі від моменту формування сигналу у датчику, через процеси оцифрування, кодування та передавання, до довготривалого зберігання або повторної обробки цифрової інформації. Наприклад, оптичні сенсори камер реєструють тепловий шум, квантові флуктуації та електричні перешкоди, які накладаються на корисний сигнал. Такі спотворення зазвичай не можуть бути повністю усунені апаратними методами, що вимагає застосування алгоритмічних технік їх пригнічення.

### Поняття та природа шуму

Шум визначається як випадкова складова сигналу, яка не несе корисної інформації та знижує точність відтворення реальних характеристик об'єкта або сцени. Його вплив може проявлятися у зміні інтенсивності пікселів, виникненні артефактів, спотворенні контурів, зменшенні контрастності, втраті текстур або загальному зниженні чіткості. У багатьох задачах комп'ютерного зору навіть незначний рівень шуму здатен значно погіршити роботу алгоритмів класифікації, сегментації чи відновлення.

Залежно від джерела та статистичних властивостей шум може належати до одного з кількох основних типів.

Найпоширенішим є гаусівський шум, який моделюється нормальним розподілом із нульовим середнім значенням. Такий шум виникає через теплові

флуктуації, електричні перешкоди та неточності вимірювань у фотодіодах сенсорів. Його особливістю є рівномірне проявлення по всій площині зображення.

Іншим поширеним видом є імпульсний шум, або «сіль і перець» (див.рис.1.1).

Він виникає у результаті збоїв у каналах зв'язку, помилок передавання або пошкодження пам'яті та проявляється у вигляді поодиноких пікселів, що набувають мінімального або максимального значення яскравості. Такий шум є надзвичайно руйнівним для локальних текстур і структури зображення.



Рис. 1.1 Приклад імпульсного шуму «сіль і перець».

Особливий інтерес становить спекл-шум, який має мультиплікативну природу. Він характерний для радарних, ультразвукових та лазерних систем візуалізації, де сигнал формується внаслідок інтерференції хвиль. Спекл-шум не додається «поверх» зображення, а модифікує його структуру, що робить його складним для традиційних методів фільтрації[1,3,4].

У реальних задачах шум нерідко має комбінований характер, поєднуючи кілька типів одночасно. Наприклад, у камерах смартфонів одночасно присутні фотонний шум, електричні флуктуації та компресійні артефакти, що значно ускладнює процес відновлення.

Вплив шуму на якість даних

Наявність шуму зменшує інформативність зображень та ускладнює роботу алгоритмів комп'ютерного зору. Шум знижує контрастність, руйнує локальні

структурні ознаки, ускладнює виділення контурів, а також може суттєво спотворювати ключові характеристики, що використовуються нейронними мережами для класифікації. Дослідження показують, що навіть слабкий гаусівський шум може знизити точність класифікації глибоких нейронних мереж на 20–40% у разі відсутності попереднього очищення даних[5].

Шум також погіршує результати об'єктного розпізнавання, методів відстеження, реконструкції 3D-сцен, медичної діагностики та інших галузей, де точність відтворення структури має критичне значення.

Ключове завдання процесу реконструкції полягає у тому, щоб максимально зменшити вплив шуму, не спотворюючи структури та текстур, які є важливими для подальшого аналізу. Проблема полягає в тому, що надмірна фільтрація може призвести до так званого «перегладжування», коли разом із шумом видаляються тонкі деталі, важливі для розпізнавання. Тому необхідно знайти баланс між подавленням шуму та збереженням корисної інформації.

Метрики оцінювання впливу шуму та якості реконструкції

Оцінювання якості очищеного зображення є невід'ємним етапом аналізу методів денойзингу. Для цього використовують кілька основних метрик, які відображають різні аспекти якості.

Однією з найпростіших і найпоширеніших метрик є MSE (Mean Squared Error) — середньоквадратична похибка. Вона визначає середнє квадратичне відхилення інтенсивності пікселів реконструйованого зображення від оригінального. Невелике значення MSE свідчить про точну реконструкцію. Однак MSE не враховує особливості людського сприйняття і не розрізняє структурних спотворень, тому її застосування до складних зображень має обмеження.

Метрика PSNR (Peak Signal-to-Noise Ratio) доповнює MSE, виражаючи відношення максимальної можливої потужності сигналу до потужності шуму. Значення PSNR вимірюється у децибелах і широко використовується для оцінки методів стискання та фільтрації. Високі значення PSNR (понад 30–40 дБ) свідчать про хорошу візуальну якість реконструкції. Проте PSNR так само не враховує

структурну подібність і тому не завжди відповідає реальному візуальному сприйняттю.

Найбільш інформативною сучасною метрикою вважається SSIM (Structural Similarity Index), розроблена з урахуванням особливостей людського зору.

SSIM оцінює подібність зображень за трьома критеріями: яскравістю, контрастністю та структурою. Це дозволяє моделі точніше відображати збереження контурів та текстур, що є ключовим показником для задач денойзингу [1,6].

### Необхідність сучасних методів відновлення

Класичні методи фільтрації, такі як середньозгладжувальні, гаусівські чи медіанні фільтри, ефективні лише за невисоких рівнів шуму та не здатні моделювати складні нелінійні залежності. Вони усувають шум переважно локально, що призводить до втрати деталей і структурної спотвореності. Зі збільшенням різноманітності даних та рівня шуму виникла потреба у більш потужних методах, які здатні враховувати глобальні та контекстні залежності у зображеннях.

Саме тому сучасні дослідження зміщуються у напрямку моделей глибинного навчання, зокрема варіаційних автокодувальників, здатних формувати латентні представлення, які розділяють шум і корисний сигнал, що забезпечує високу точність реконструкції навіть у складних умовах.

## **1.2. Методи фільтрації та попереднього очищення даних.**

Перші методи до зменшення шуму ґрунтувалися на лінійних фільтрах, які усувають високочастотні компоненти сигналу.

Гаусівський фільтр [15]:

Гаусівський фільтр є одним з найпоширеніших методів фільтрації зображень для зменшення шуму. Він використовує гаусівське згладжування, що означає згладжування зображення за допомогою гаусівської функції. Цей метод ефективно видаляє шум, але також може призвести до розмивання країв і деталей зображення.

Принцип роботи Гаусівського фільтру:

Гаусівський фільтр застосовує згладжування зображення за допомогою гаусівської функції розподілу, яка визначає, наскільки сильно впливатимуть сусідні пікселі на значення центрального пікселя. Основна ідея полягає в тому, щоб усереднити пікселі зважено, причому вага визначається гаусівським розподілом.

#### Кроки застосування Гаусівського фільтру

Створення Гаусівського ядра: Першим кроком є створення матриці (ядра), що містить гаусівські коефіцієнти. Розмір ядра зазвичай обирається непарним (наприклад, 3x3, 5x5, 7x7), а значення в ядрі залежать від стандартного відхилення  $\sigma$ .

Конволюція зображення з ядром: Ядро застосовується до зображення через операцію згортки (конволюції). Це означає, що кожен піксель у зображенні замінюється зваженим середнім значенням пікселів у його локальному сусідстві, де ваги визначаються гаусівським ядром.

#### Переваги:

Простота і ефективність: Гаусівський фільтр легко реалізувати і використовувати. Він ефективно зменшує гаусівський шум, особливо коли шум розподілений рівномірно по всьому зображенню. Гладкість результату: Гаусівське згладжування забезпечує плавне і природне вигляд згладжених областей.

#### Недоліки:

Розмиття країв: Головний недолік Гаусівського фільтру – розмивання країв та деталей зображення, що може призвести до втрати важливої інформації. Однорідність згладжування: Гаусівський фільтр застосовує однакове згладжування до всього зображення, незалежно від локальних особливостей, таких як краї та текстури.



Рис. 1.2 Приклад гаусівського фільтру

Білатеральний фільтр[16]:

Білатеральний фільтр (Bilateral Filter) є популярним методом фільтрації зображень, який зберігає краї, виконуючи згладжування з урахуванням як просторового положення пікселів, так і їх інтенсивності. Це робить його ефективним для придушення шуму, зберігаючи при цьому текстури і деталі зображення.

Принцип роботи Білатерального фільтру

Білатеральний фільтр використовує дві вагові функції для визначення впливу сусідніх пікселів на значення центрального пікселя: одна функція враховує просторове положення пікселів, а інша – різницю в інтенсивності пікселів.

Функція просторового зважування Просторове зважування виконується за допомогою гаусівської функції, яка визначає вплив сусідніх пікселів залежно від їх відстані до центрального пікселя. Визначається за формулою (1.1).

$$G_s(x, y) = \exp\left(-\frac{x^2+y^2}{2\sigma_s^2}\right) \quad (1.1)$$

Функція інтенсивності зважування Інтенсивність зважування визначається за допомогою гаусівської функції, яка визначає вплив сусідніх пікселів залежно від різниці їх інтенсивності з центральним пікселем. Визначається за формулою (1.2).

$$G_r(I, J) = \exp\left(-\frac{(I-J)^2}{2\sigma_g^2}\right) \quad (1.2)$$

Згортка Білатерального фільтра Формула для Білатерального фільтра визначається за формулою (1.3).

$$I'(p) = \frac{1}{W_p} \sum_{q \in S} I(q) \cdot G_s(\|p - q\|) \cdot G_r(|I(p) - I(q)|) \quad (1.3)$$

де – інтенсивність центрального пікселя, – інтенсивність сусіднього пікселя, – гаусівська функція просторового зважування, – гаусівська функція зважування за інтенсивністю, – нормуючий коефіцієнт, що забезпечує збереження яскравості, що визначається за формулою (1.4).

$$W_p = \sum_{q \in S} G_s(\|p - q\|) \cdot G_r(|I(p) - I(q)|) \quad (1.4)$$

Вибір параметрів фільтра: у формулах (3.2) та (3.3) вибираються значення для просторового зважування і для зважування за інтенсивністю. Значення визначає, наскільки сильно враховується відстань між пікселями, а – наскільки сильно враховується різниця в інтенсивності.

Застосування згортки: для кожного пікселя  $p$  обчислюється нове значення, враховуючи вплив сусідніх пікселів  $q$  з урахуванням як їх просторового положення, так і різниці в інтенсивності.

Переваги та недоліки

Переваги: Збереження країв: Завдяки зважуванню за інтенсивністю, фільтр зберігає різкі краї і деталі, що робить його ідеальним для текстурованих зображень.

Ефективне придушення шуму: Білатеральний фільтр добре зменшує шум, особливо гаусівський шум.

Недоліки: Обчислювальна складність: Білатеральний фільтр є більш обчислювально затратним порівняно з традиційними фільтрами, такий як гаусівський фільтр, через необхідність обчислення ваг для кожного пікселя в локальному вікні. Параметри фільтра: Вибір відповідних значень  $\epsilon$  є критичним для досягнення бажаного балансу між згладжуванням і збереженням деталей.

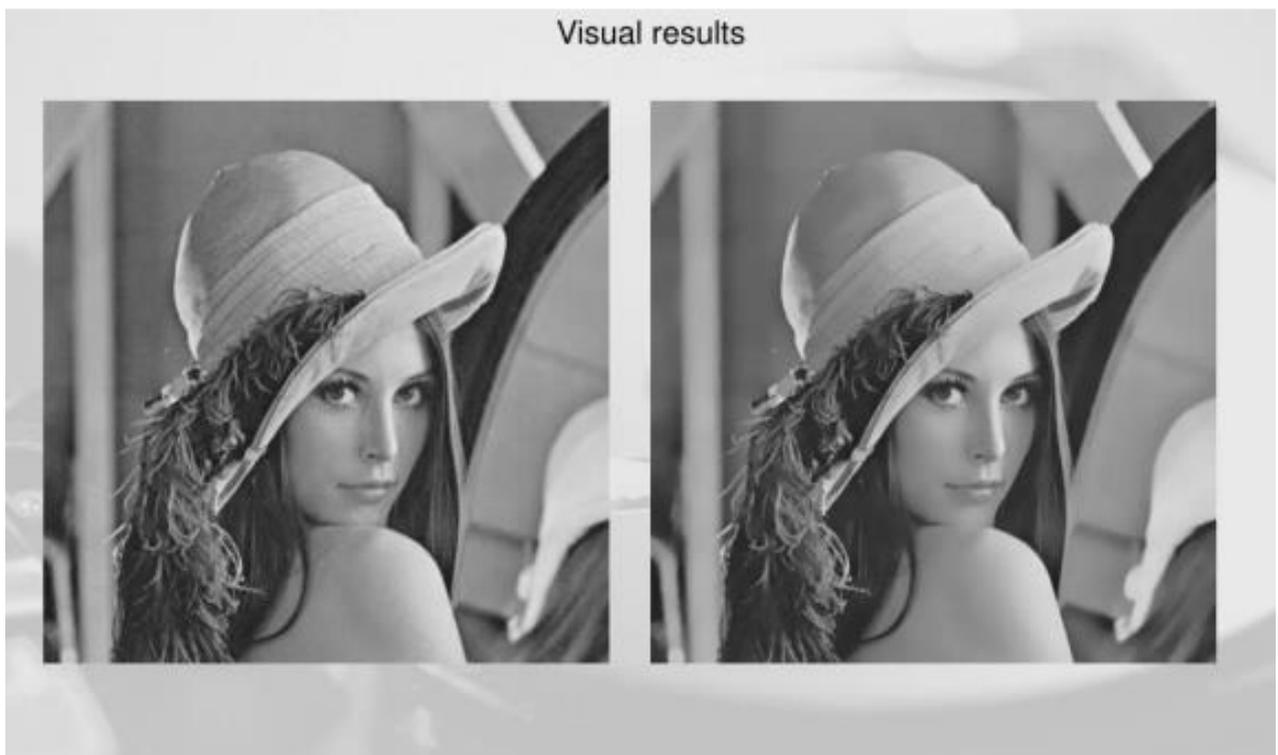


Рис. 1.3 Приклад Білатерального фільтру

Перетворення Вейвлет[17]:

Перетворення вейвлет (Wavelet Transform) є потужним інструментом в обробці зображень, яке дозволяє розкласти зображення на різні частотні компоненти. Це дозволяє ефективно відокремлювати шум, який зазвичай міститься у високочастотних компонентах, і зберігати важливу інформацію у низькочастотних компонентах. Розглянемо детальніше принцип роботи цього методу, його переваги та недоліки.

Принцип роботи перетворення вейвлет

## Розклад зображення

Перетворення вейвлетів виконує розклад зображення на набори компонентів з різними частотами та просторовими локалізаціями. Це здійснюється за допомогою вейвлет-функцій, які аналізують зображення на різних масштабах.

Низькочастотні компоненти (LL): зберігають загальну структуру та основну інформацію зображення.

Високочастотні компоненти (LH, HL, HH): містять інформацію про деталі та краї зображення. Шум зазвичай присутній у цих компонентах.

## Порогування

Після розкладу зображення на вейвлет-компоненти, високочастотні компоненти піддаються порогованню для зменшення шуму. Порогування може бути жорстким (hard) або м'яким (soft):

Жорстке порогоування: значення, менші за певний поріг, обнуляються.

М'яке порогоування: значення, менші за поріг, обнуляються, а значення, що перевищують поріг, зменшуються на величину порогу. Переваги перетворення вейвлет

Збереження деталей: вейвлети дозволяють зберігати важливі деталі зображення при зменшенні шуму, оскільки вони працюють на різних масштабах і частотах.

Ефективне зменшення шуму: вейвлети добре справляються з гаусівським шумом та іншими типами шуму, що присутні у високочастотних компонентах.

Адаптивність: перетворення вейвлетів адаптується до локальних особливостей зображення, що дозволяє більш точно розрізнити шум і корисну інформацію.

## Недоліки перетворення вейвлет

Обчислювальна складність: розрахунок вейвлет-перетворення і подальше порогоування можуть бути обчислювально затратними, особливо для великих зображень.

Вибір вейвлет-функції: якість обробки залежить від вибору відповідної вейвлет-функції та параметрів порогування. Неправильний вибір може призвести до втрати деталей або недостатнього придушення шуму.

Граничні ефекти: при обробці країв зображення можуть виникати артефакти, якщо обрані методи екстраполяції неадекватні.

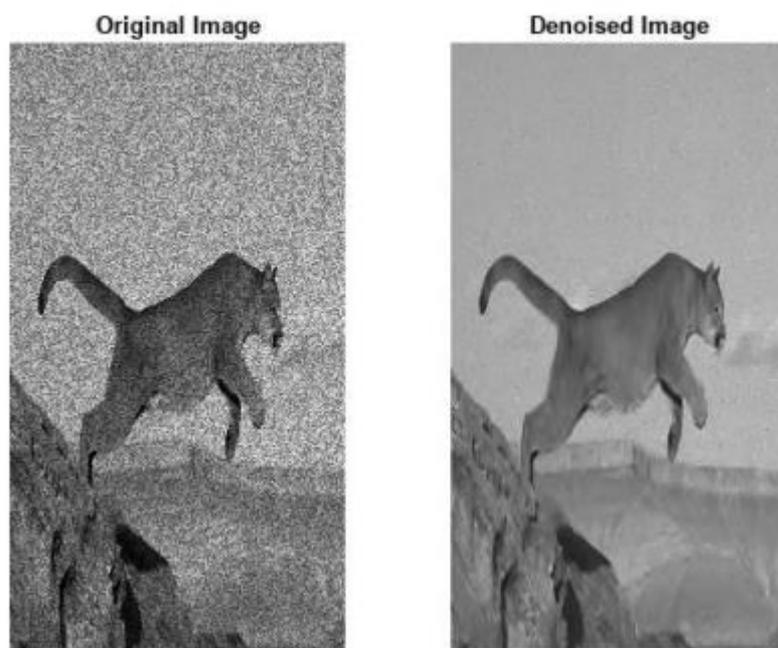


Рис. 1.4 Приклад Вейльвет[18]

### 1.3. Традиційні та сучасні підходи до реконструкції сигналів і зображень.

Проблема реконструкції сигналів і зображень у присутності шуму є однією з центральних у цифровій обробці інформації та комп'ютерному зорі. Ефективне відновлення даних є критичним для різних сфер — від медичної візуалізації та супутникових систем спостереження до побутових камер і промислового контролю. У різні епохи розвитку інформаційних технологій застосовувалися різні підходи до реконструкції, починаючи з класичних лінійних методів до сучасних моделей глибокого навчання.

У цьому підрозділі розглядаються основні етапи еволюції методів відновлення сигналів та зображень з акцентом на їх можливості, обмеження та передумови для появи глибинних архітектур, таких як варіаційний автокодувальник.

### Лінійні методи та регресійні моделі

На ранніх етапах розвитку цифрової обробки сигналів домінували лінійні алгоритми, що базувалися на припущеннях про адитивну природу шуму та локальну неперервність сигналу. Найпоширенішими методами були лінійна регресія, фільтр найменших квадратів (LS), регресія з регуляризацією (ridge, LASSO), які націлювалися на пошук оптимального лінійного відображення між зашумленим сигналом і його очікуваним «чистим» аналогом. Такі моделі застосовувалися переважно в задачах згладжування, прогнозування та базового відновлення спотворених даних.

Попри їхню математичну простоту, лінійні методи мають суттєві обмеження. Вони не здатні моделювати складні нелінійні залежності, не враховують локальні та глобальні структурні характеристики зображення й погано працюють у випадках, коли шум порушує статистичні припущення моделі. Наприклад, гаусівський шум ще може бути ефективно пригнічений лінійними оцінками, але імпульсний, спекл- або комбінований шум вимагає складніших, нелінійних підходів.

### Лінійно-фільтраційні методи реконструкції

У галузі цифрової обробки зображень широко застосовувалися традиційні фільтри: середньозгладжувальний, гаусівський, вейвлет-фільтри, медіанний фільтр і низка інших.

Гаусівський фільтр ефективний для низькочастотного шуму, але викликає розмиття зображення та втрату контурів.

Медіанний фільтр добре працює з імпульсним шумом, але часто деформує дрібні деталі.

Анізотропна дифузія та нелокальні методи (наприклад, Non-Local Means) значно підвищили якість відновлення, однак їхня обчислювальна складність та чутливість до параметрів обмежували їх широке застосування.

Загальним недоліком традиційних фільтраційних технік є те, що вони ґрунтуються на жорстких припущеннях про природу шуму та структуру сигналу. Вони також не здатні враховувати високорівневі семантичні залежності, які є ключовими при аналізі складних зображень, таких як обличчя, медичні томографії або природні сцени.

Перехід до машинного навчання: перші нейронні моделі

Поява машинного навчання стала важливим етапом у розвитку методів відновлення зображень. Регресійні моделі та базові нейронні мережі, такі як багатошаровий перцептрон (MLP), почали використовуватися як універсальні апроксиматори нелінійних залежностей. MLP міг навчатися знаходити складні функції відображення між зашумленими та чистими зображеннями, що забезпечувало значно кращу якість реконструкції порівняно з лінійними моделями.

Однак нейронні мережі простих архітектур мали низку недоліків:

- вони опрацьовували дані у векторному вигляді, що руйнувало просторову структуру зображення;
- вони не враховували локальні кореляції між сусідніми пікселями;
- збільшення розміру вхідних даних приводило до різкого росту кількості параметрів.

Через це базові нейронні моделі не могли повноцінно конкурувати з методами обробки зображень, поки не було створено новий тип мереж — згорткові нейронні мережі (CNN).

Згорткові нейронні мережі та революція в обробці зображень

З розвитком глибинного навчання й появою CNN відбулася справжня революція в задачах реконструкції та очищення зображень. Згорткові мережі враховують просторову структуру зображення та здатні витягувати високорівневі ознаки, які традиційні фільтри не могли врахувати.

CNN-моделі для денойзингу, такі як DnCNN, FFDNet, IRCNN, досягли значних успіхів у видаленні гаусівського шуму, а також у боротьбі зі складнішими типами спотворень.

Однак, незважаючи на високу точність, вони мають певні обмеження: CNN моделюють залежності локально, обмежуючись рецептивним полем, вони не оперують латентним простором у контексті ймовірнісної моделі, їхня здатність до генерації та відновлення складних патернів є обмеженою. Саме ці недоліки стали мотиватором для розвитку генеративних моделей — глибших, більш узагальнювальних і здатних моделювати розподіли даних.

### Поява генеративних моделей та варіаційних автокодувальників

Генеративні моделі, такі як Variational Autoencoder (VAE) та Generative Adversarial Networks (GAN), відкрили нові можливості у сфері реконструкції зображень.

VAE запропонував принципово новий підхід — моделювання латентного простору як ймовірнісного розподілу, що дозволяє більш повно враховувати варіабельність та структуру даних.

Наявність латентного простору дозволяє VAE:

- розділяти шум і корисний сигнал на різні компоненти;
- формувати гладкий та впорядкований латентний розподіл;
- ефективно реконструювати навіть сильно спотворені зображення;
- генерувати нові варіації зображень;

У задачах денойзингу VAE особливо цінні тим, що вони не просто фільтрують шум, а навчаються статистичній моделі чистих зображень, що дозволяє відновлювати дані навіть тоді, коли шум не відповідає типовим моделям.

Сучасні підходи: гібридні, варіаційні та трансформерні моделі

Останні роки ознаменувалися появою нових, ще потужніших методів реконструкції зображень.

До них належать:

- VAE з поділом латентного простору на семантичний і шумовий компоненти;
- гібридні VAE-GAN моделі;
- дифузійні моделі для відновлення даних;
- трансформерні моделі, які працюють із зображеннями як з послідовністю патчів.

Усі ці підходи демонструють значно вищу здатність моделювати нелінійні спотворення, зберігати структури зображення та оцінювати достовірність реконструкції.

## 1.4 Автокодувальники

Проблема відновлення зашумленої інформації є однією з фундаментальних у галузях цифрової обробки сигналів, комп'ютерного зору та машинного навчання. Джерелами шуму можуть бути фізичні сенсори, недосконалість каналів передачі, низька освітленість або обмежені технічні параметри обладнання. Тому завдання реконструкції чистого сигналу з його спотвореної версії перетворилося на важливу складову сучасних інтелектуальних систем. За останні десятиріччя у цій сфері відбувся значний прогрес — від простих лінійних моделей до складних глибинних нейронних архітектур, що дозволяють моделювати розподіли даних і враховувати нелінійні залежності. Серед таких методів особливий інтерес становлять автокодувальники та їх модифікації, які стали основою багатьох сучасних моделей генерації та реконструкції.

Автокодувальник як базова архітектура нелінійного стискання

Автокодувальник (Autoencoder, AE) — це тип штучної нейронної мережі, що складається з двох компонентів: енкодера та декодера. Мета моделі полягає у тому, щоб навчитися стискати вихідні дані у вектор меншої розмірності, а потім реконструювати їх якнайточніше. Вхідні дані подаються на енкодер, який формує компактне представлення (латентний вектор), що зберігає найбільш інформативні

характеристики сигналу. Декодер, у свою чергу, відновлює вихідний зразок на основі цієї компактної репрезентації[9].

Навчання автокодувальника відбувається в режимі неконтрольованого навчання: модель порівнює вхідні дані та їх реконструкцію й намагається мінімізувати різницю між ними. Цільова функція, як правило, включає різницю між оригінальним зображенням та відновленим результатом. Зазвичай використовують середньоквадратичну або абсолютну похибку.

Однією з ключових переваг автокодувальників є їх здатність виділяти приховані, суттєві ознаки даних. Латентний простір, сформований моделлю, часто містить стислий, але інформативний опис вхідного сигналу. Це дає можливість відновити дані навіть за часткової втрати інформації, а також застосовувати автокодувальники для кластеризації, зменшення розмірності та аномалійного аналізу.

Проте класичний автокодувальник не враховує специфіки шуму. Оскільки його навчальна мета полягає у відтворенні вхідних даних якомога точніше, модель копіює ці дані незалежно від наявності спотворень. Таким чином, у випадках зашумлених зображень автокодувальник не має механізму для відокремлення шуму від корисного сигналу — латентний простір змішує ці компоненти, що погіршує якість реконструкції.

Denoising Autoencoder (DAE) — розширення автокодувальника для очищення даних (див. рис 1.5)

Для розв'язання проблеми роботи з шумом було запропоновано модифікацію — денойзинговий автокодувальник (Denoising Autoencoder, DAE). Його ключова відмінність від традиційного АЕ полягає в тому, що він спеціально навчається відновлювати чистий сигнал з зашумленого вхідного зразка. На тренувальному етапі моделі подається спотворене зображення, тоді як ціллю є чистий зразок. Таким чином, модель явно вчиться відділяти шумові компоненти від структурних елементів вхідних даних.

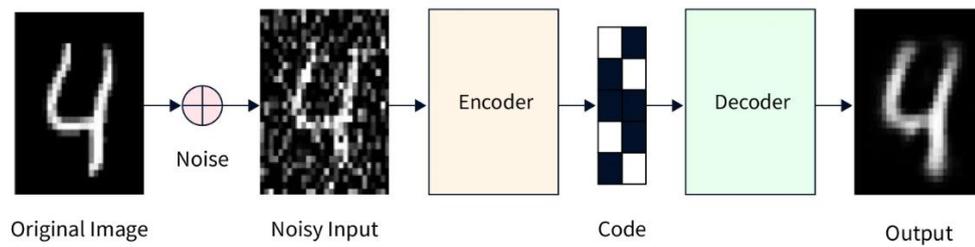


Рис. 1.5 Приклад роботи денойзингового автокодувальника

DAE демонструє значно кращу ефективність при обробці випадкових спотворень, оскільки він навчається розпізнавати закономірності шуму та відновлювати природні структури зображення. Крім того, DAE здатний узагальнювати знання на різні рівні шуму, що робить його ефективним для широкого спектра сигналів: зображень, аудіоданих, часових рядів, датчиків.

Втім, попри переваги, денойзинговий автокодувальник має низку фундаментальних обмежень. Найважливішим є відсутність потенційної ймовірнісної інтерпретації. Латентний простір формується детерміновано, без урахування статистичних властивостей даних. Це обмежує генеративні можливості DAE та робить його менш стійким до шумів, які відрізняються від тих, на яких модель була попередньо навчена. Крім того, DAE не може формувати варіативні реконструкції та не дає можливості контролювати розподіл латентного представлення.

#### Згорткові нейронні мережі (CNN) у задачах відновлення зображень

Поява згорткових нейронних мереж (CNN) докорінно змінила підходи до реконструкції даних. CNN-підходи стали фундаментом багатьох сучасних алгоритмів денойзингу завдяки здатності ефективно узагальнювати локальні та глобальні характеристики зображення.

Головним елементом CNN є згорткові фільтри, що виявляють характерні патерни: краї, текстури, повторювані структури. Це робить CNN особливо ефективними при роботі з природними зображеннями, де наявні складні контрасти та просторові кореляції.

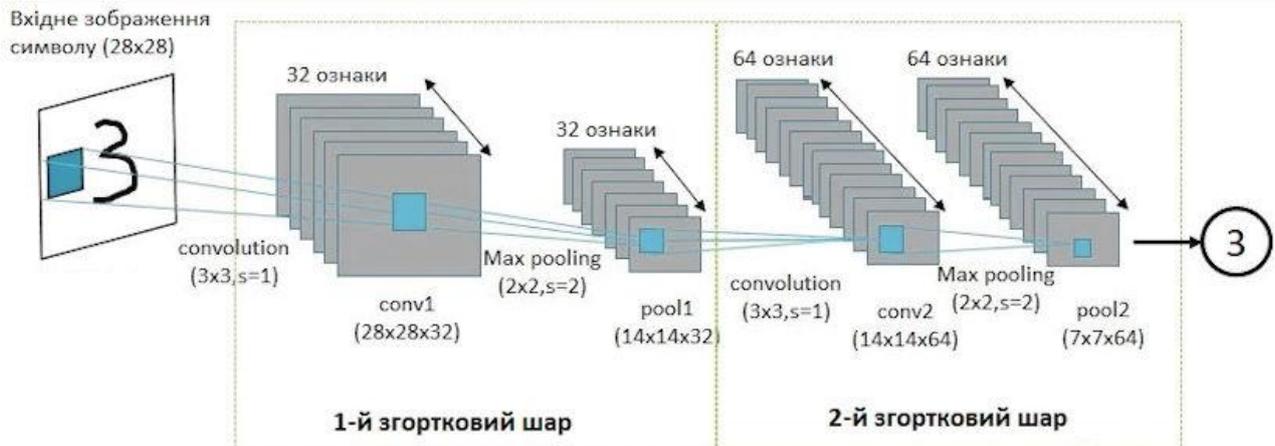


Рис. 1.6 Згорткові нейронні мережі

До найвідоміших моделей відновлення зображень належать:

DnCNN (Denoising CNN) — одна з перших глибоких мереж, яка навчалася не напряму відновлювати чисте зображення, а прогнозувати шум, що значно покращило стабільність моделі;

RED-Net — архітектура з прямими пропусковими з'єднаннями між симетричними шарами енкодера та декодера, що дозволяє уникати втрати інформації при глибоких мережах;

U-Net, яка завдяки текстовим skip-з'єднанням здатна ефективно зберігати локальні структури та добре справляється зі складними шумами.

Попри їх безперечні переваги, CNN-підходи також мають суттєві недоліки.

По-перше, вони значною мірою залежать від конкретного типу шуму. Мережа може втрачати ефективність, якщо характер шуму змінюється.

По-друге, CNN не будують ймовірнісної моделі даних, а лише емпірично вчаться відновлювати структури. Це обмежує їхню здатність до узагальнення та робить їх менш стійкими до нестандартних спотворень. Також CNN-підходи вимагають значних обчислювальних ресурсів, особливо у випадках глибоких моделей із широкими фільтрами та багатьма параметрами.

## 1.5 Варіаційні автокодувальники

Варіаційні автокодувальники (Variational Autoencoders, VAE) виникли як природний розвиток класичних автокодувальників із метою подолання їхніх фундаментальних обмежень. Звичайні автокодувальники здійснюють детерміноване відображення вхідного прикладу у фіксований вектор латентного простору, що обмежує їх здатність моделювати складні варіації даних та описувати невизначеність. VAE розширюють класичний AE шляхом введення ймовірнісної інтерпретації латентного простору, що дає змогу моделювати розподіл прихованих змінних та виконувати повноцінний байєсівський висновок.

В основі варіаційного автокодувальника лежить припущення, що кожне вхідне зображення приховано залежить від певних латентних факторів, які описуються випадковими змінними. На відміну від детермінованого вектора, VAE представляє прихідне представлення у вигляді параметризованого розподілу, найчастіше нормального багатовимірного розподілу з певними середніми значеннями та дисперсіями. Це дозволяє не просто стискати дані, а моделювати їхню стохастичну природу.

Енкодер VAE перетворює вхідний приклад у параметри розподілу латентних змінних — зазвичай це вектор середніх значень та вектор стандартних відхилень. Таким чином, замість єдиного латентного коду формується розподіл, з якого можна вибирати різні реалізації латентних змінних. Це відображає властивість даних мати варіативність, що особливо актуально при роботі з шумами та іншими стохастичними спотвореннями.

Архітектура VAE включає ключовий механізм — перепараметризаційний трюк, який дозволяє реалізувати вибірку зі стохастичного розподілу у спосіб, сумісний із процедурою зворотного розповсюдження похибки. Завдяки цьому модель може навчатися енд-ту-енд, а вибірки латентних змінних стають диференційовними щодо параметрів мережі. Перепараметризація робить стохастичну природу латентного простору формально прозорою для оптимізації.

У процесі навчання варіаційний автокодувальник прагне розв'язати задачу максимізації правдоподібності даних. Оскільки точне обчислення правдоподібності є недосяжним через складність інтегралів за латентними змінними, застосовується

варіаційний підхід — максимізація нижньої межі правдоподібності (ELBO). Ця межа складається з двох складових: терміна реконструкції та регуляризації, яка вимірюється через дивергенцію Кульбака–Лейблера між апіорним та апроксимаційним апостеріорним розподілами латентних змінних.

Реконструкційний термін оцінює, наскільки добре декодер відтворює оригінальне зображення на основі вибраного латентного зразка. Це аналогічно базовому автокодувальнику, але з тією різницею, що реконструкція базується на вибірці зі стохастичного простору. Чим менша похибка реконструкції, тим краще модель захоплює суттєві структурні властивості даних.

Регуляризаційний компонент ELBO змушує латентний простір приймати впорядковану, гладку структуру, наближену до заданого апіорного розподілу (звичайно стандартного нормального). Дивергенція Кульбака–Лейблера виступає природним регулятором, що запобігає перенавчанню та забезпечує смислову організацію латентного простору. Саме завдяки цьому VAE уникає хаотичності латентних представлень, характерної для звичайних AE, і може узагальнювати дані на нові приклади.

У контексті задачі відновлення даних із шумом це має вирішальне значення. Стохастичний характер латентного простору дозволяє моделі «розуміти» природу випадковості у вхідних даних. VAE здатний відокремлювати закономірні структури зображення від випадкових шумів, оскільки регуляризація не дозволяє моделі просто копіювати спотворення. Впорядкований латентний простір забезпечує стійкість до нових типів спотворень, які можуть не бути представлені у навчальній вибірці.

На відміну від денойзингових автокодувальників, VAE здатний моделювати широкий клас шумових розподілів завдяки своїй імовірнісній природі. Якщо для DAE важливо, яким саме чином було додано шум під час навчання, то VAE здатний узагальнювати його властивості, оскільки латентний простір фактично описує узагальнений розподіл вхідних даних. Це робить VAE дуже ефективним для задач, де шум має складну структуру, нелінійні залежності або стохастичну природу.

Завдяки регуляризації KL-дивергенцією варіаційні автокодувальники уникають переадаптації до навчальних прикладів і формують гладку геометрію латентного простору. Природно, що на практиці це полегшує інтерполяцію між різними зразками, покращує стабільність реконструкцій та визначає коректні переходи між класами даних.

У задачах очищення зображень VAE має важливу перевагу перед звичайними CNN-підходами чи DAE: модель не тільки прибирає шум, але й дає інтерпретоване ймовірнісне представлення про те, які саме структури є типовими для даного класу зображень і як вони можуть варіюватися. Це дозволяє будувати більш стійкі моделі та адаптувати їх до різноманітних шумових режимів, включно з тими, яких не було у навчальній вибірці.

Таким чином, варіаційні автокодувальники є фундаментально важливим інструментом сучасної реконструкції даних. Вони поєднують переваги класичних автокодувальників та баєсівських методів, забезпечуючи одночасно точність реконструкції, стохастичність опису та стійкість до шуму.

Для задачі, що розглядається у даній роботі оптимізації процесу відновлення зашумлених зображень VAE створює найсприятливіші умови для моделювання як семантичних, так і шумових компонент даних(див. в рис.1.6).

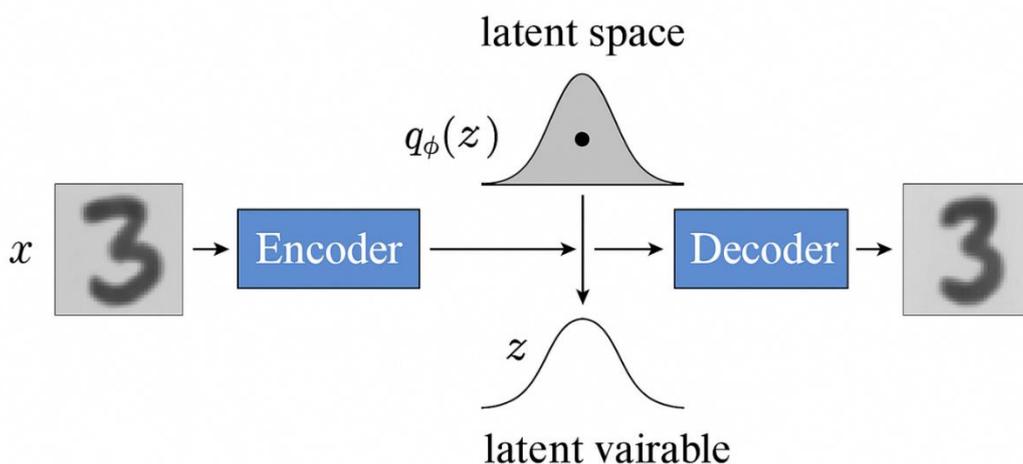


Рис. 1.6 Варіаційний атокодувальник

Мета моделі — навчитися максимально наближеною до істинного апостеріорного розподілу  $q_{\theta} = (z|x)$  функцією.

Оптимізація виконується шляхом максимізації нижньої межі правдоподібності (ELBO)(1.5):

$$\mathcal{L}_{VAE} = -E_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] + K L(q_{\phi}(z|x)||p(z)) \quad (1.5)$$

$KL(\cdot||\cdot)$  — дивергенція Кульбака–Лейблера.

Це дозволяє VAE не лише реконструювати дані, але й моделювати ймовірнісну природу шуму, що робить їх особливо придатними для задач відновлення даних із випадковими спотвореннями[11, 14].

Завдяки регуляризації KL-дивергенцією модель уникає перенавчання, а латентний простір стає гладким і структурованим. розробці методів відновлення даних.

Таблиця 1.1

## Оцінка ефективності методів

| Метод            | Тип моделі           | Основна ідея                                      | Переваги                                   | Недоліки  |
|------------------|----------------------|---|--|---|
| Класичні фільтри | Лінійна              | Усереднення або згладжування сигналу              | Простота, швидкість                        | Втрата деталей, неадаптивність                        |
| DAE              | Нейрона              | Навчання відновленню «чистих» даних із зашумлених | Хороша якість реконструкції                | Детермінований латентний простір                      |
| CNN              | Нейронна (згорткова) | Просторове фільтрування та навчання на патернах   | Висока точність, стабільність              | Висока складність, залежність від типу шуму           |
| VAE              | Імовірнісна нейронна | тохастичне відновлення та генерація даних         | Узагальнення, контроль латентного простору | Складність навчання, можлива розмитість реконструкцій |

## 2 МЕТОДОЛОГІЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ВІДНОВЛЕННЯ ДАНИХ НА ОСНОВІ VAE

### 2.1. Постановка задачі та формальний опис моделі.

В умовах сучасних систем комп'ютерного зору та обробки цифрових зображень значна частина даних зазнає спотворення, спричиненого різноманітними типами шуму: сенсорним, імпульсним, гаусівським, текстурним або комбінованим. Наявність шуму ускладнює подальший аналіз інформації, знижує точність класифікації, сегментації, розпізнавання та суттєво погіршує візуальну якість зображень.

Тому виникає необхідність у створенні моделей, здатних ефективно відновлювати зашумлені дані, зберігаючи при цьому структуру, контури та семантичний зміст.

Класичні методи фільтрації демонструють обмежені можливості, оскільки вони або надто розмивають краї (лінійні фільтри), або працюють лише з певними типами шуму (медіанний фільтр). Сучасні глибокі нейронні мережі, зокрема автоенкодера та згорткові моделі, значно покращили якість реконструкції, проте їх латентний простір часто змішує семантичні та шумові компоненти, що знижує стійкість до різних видів спотворень.

Варіаційний автокодувальник (VAE) є перспективним інструментом для створення імовірнісної моделі даних, проте базова архітектура VAE не завжди дозволяє забезпечити чітке розділення між корисним сигналом та шумом. Це обмежує точність відновлення, особливо при складних або нестандартних типах шуму. У цьому контексті актуальною стає задача оптимізації архітектури VAE шляхом модифікації латентного простору та функції втрат.

Основна задача дослідження полягає у розробленні моделі, здатної:

1. Ефективно відокремлювати шумові компоненти від змістових ознак зображення;

2. Формувати структурований латентний простір;
3. Забезпечувати стабільне та точне відновлення зображень у присутності різних типів шуму;
4. Оптимізувати процес реконструкції шляхом вдосконалення архітектури та функції втрат варіаційного автокодувальника.

#### Формальний опис моделі

Розроблювана модель базується на модифікованій архітектурі варіаційного автокодувальника, у якій латентний простір поділено на дві незалежні компоненти:

Семантичний латентний підпростір — відповідає за зміст зображення, його структуру, контури та основні ознаки;

Шумовий латентний підпростір — моделює інтенсивність, тип та просторові властивості шуму.

Такий підхід дозволяє моделі явно розділити інформацію про зображення та шум, що підвищує якість реконструкції.

## **2.2. Архітектура варіаційного автокодувальника для роботи із зашумленими даними.**

Архітектура варіаційного автокодувальника (VAE) є однією з найважливіших концепцій сучасних методів глибокого навчання, заснованих на варіаційному баєсовому підході та ймовірнісному моделюванні латентних просторів. Незважаючи на схожість зі стандартними автокодувальниками, варіаційні автокодувальники мають принципово іншу математичну основу, яка дозволяє не просто стискати інформацію, але й формувати структурований латентний простір із чітко визначеними статистичними властивостями. Саме ця особливість робить VAE надзвичайно ефективним у задачах відновлення зображень, зокрема при роботі з даними, спотвореними різними типами шуму.

Варіаційний автокодувальник складається з двох основних компонентів — енкодера та декодера.

Енкодер є штучною нейронною мережею, що отримує на вході зображення великої розмірності та трансформує його у компактне латентне представлення. На відміну від класичного автокодувальника, який формує єдиний детермінований вектор, VAE будує дві окремі параметризації латентного простору: вектор середніх значень  $\mu(x)$  і вектор логарифмів дисперсій  $\sigma^2(x)$ . Ця ймовірнісна параметризація дозволяє інтерпретувати латентне представлення як вибірку із багатовимірного нормального розподілу, що відкриває можливість моделювання даних та генерації нових їх варіацій.

Природа латентного простору є фундаментальним елементом VAE. Вона забезпечує безперервність, гладкість та впорядкованість розподілу, що є надзвичайно важливим для задач денойзингу. У стандартних автокодувальниках латентний простір може мати хаотичну форму, у ньому можливі «порожні області» та відсутність структурної відповідності між подібними зображеннями.

У VAE функції втрат, який вимірюється через дивергенцію Кульбака–Лейблера, сприяє вирівнюванню розподілу до багатовимірного розподілу нормального типу. Це гарантує, що різні зашумлені входи будуть проектуватися у впорядковані області латентного простору, що значно підвищує якість відновлення.

Одним із ключових технічних аспектів архітектури є застосування перепараметризаційного трюку (reparameterization trick), який дозволяє обчислювати градієнти через стохастичний процес.

Замість прямої вибірки  $z \sim N(\mu, \sigma^2)$

модель формує  $z = \mu + \sigma \odot \varepsilon$ ,

де  $\varepsilon$  — вибірка зі стандартного нормального розподілу.

Така трансформація забезпечує можливість застосування алгоритму зворотного поширення похибки та стабільного навчання всієї мережі. Саме це дозволяє VAE працювати як класичну нейронну архітектуру, не порушуючи при цьому її ймовірнісної природи.

Декодер у VAE виконує роль зворотного перетворення. Він приймає вибірку з латентного простору та реконструює зображення, максимально наближене до оригінального. Якщо в класичному автокодувальнику декодер працює суто

детерміновано, то у VAE його завданням є реконструювати очікуване значення умовного розподілу  $p(x|z)$ , що дозволяє йому бути більш гнучким та стійким до шумових спотворень. У задачах денойзингу це надважливо, адже модель фактично навчається знаходити «чисту» структуру навіть у випадках, коли вхідне зображення суттєво спотворене шумом.

Особливо важливою є модифікація архітектури VAE для роботи з зашумленими даними. Замість традиційного підходу, коли модель навчається відтворювати вхід, у задачі денойзингу декодер прагне реконструювати чисте зображення  $x$ , тоді як на вхід енкодера подається шумова версія  $y$ . Такий метод називається denoising VAE і дозволяє моделі навчитися відділяти змістовні характеристики зображення від випадкових шумових компонентів.

У роботі застосовується варіант — поділ латентного простору на два підкомпоненти: семантичний ( $z_{\text{semantic}}$ ), що відповідає за структурну інформацію зображення, і шумовий ( $z_{\text{noise}}$ ), який моделює випадкові спотворення. Такий підхід забезпечує ще більш точне відокремлення сигналу від шуму, підсилюючи здатність моделі очищувати зображення без втрати деталей.

У контексті відновлення зашумлених даних варіаційний автокодувальник демонструє важливі переваги. Його латентний простір формується не лише для стискування, але й для моделювання статистичної природи шуму. Це дозволяє моделі працювати навіть з незнайомими типами спотворень, оскільки регуляризований латентний простір запобігає перенавчанню на конкретний тип шуму та сприяє узагальненню. У той час як класичні автокодувальники часто зазнають труднощів при роботі з високочастотним або структурованим шумом, варіаційні моделі демонструють здатність до більш глибокого аналізу патернів та точного їх розділення.

Функція втрат у VAE також зазнає значної модифікації порівняно зі звичайними автокодувальниками. Вона складається з двох компонентів: реконструкційної (яка мінімізує помилку між відновленим та чистим зображенням) та регуляризаційної (KL-дивергенції), що контролює форму латентного простору. У денойзингу застосовують додаткові складові, які включають структурні втрати

(SSIM-loss), перцептивні втрати або штрафи за шумові компоненти. Така система дозволяє значно покращити здатність моделі зберігати важливі деталі, усуваючи при цьому шум.

У сукупності архітектура варіаційного автокодувальника Рис.(2.1) створює гнучку платформу для моделювання та реконструкції зашумлених даних. Наявність ймовірнісного латентного простору, перепараметризаційного трюку, регуляризаційної складової та можливості модифікації структури дозволяє адаптувати VAE для широкого спектра задач, пов'язаних із очисткою зображень, включаючи медичні застосування, супутникову обробку даних та відновлення фотографій, отриманих у складних умовах.

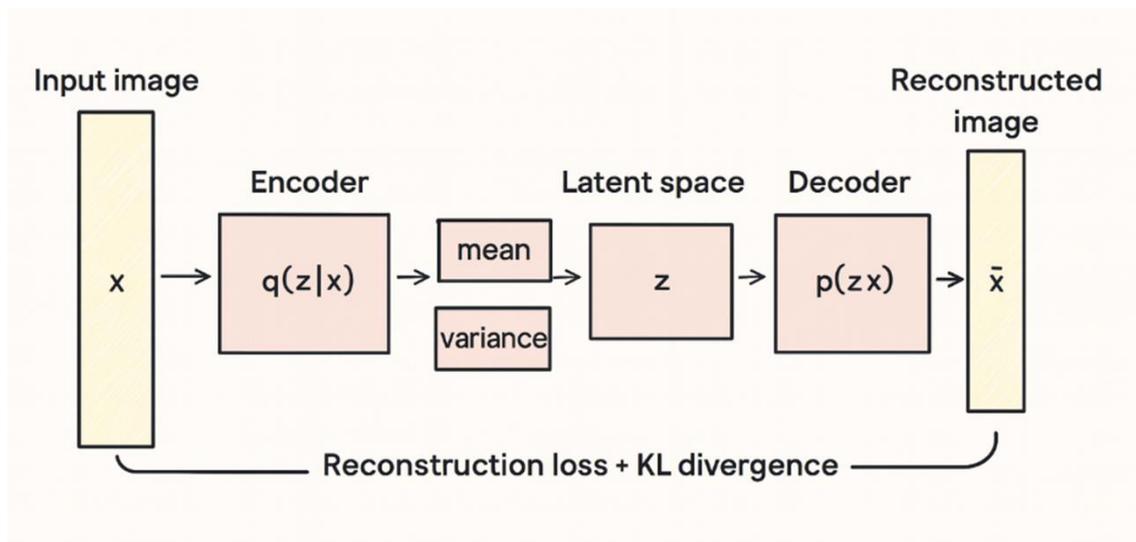


Рис. 2.1 Архітектура VAE

### 2.3. Методи оптимізації.

Оптимізація варіаційного автокодувальника є комплексним процесом, який охоплює кілька взаємопов'язаних аспектів: регуляризацію латентного простору, налаштування гіперпараметрів моделі та коректний вибір функції втрат. Саме поєднання цих підходів забезпечує стабільне навчання, якісну реконструкцію та стійкість до шумів, що особливо важливо у задачах відновлення цифрових зображень.

## Регуляризація VAE

Регуляризація відіграє ключову роль у формуванні впорядкованого латентного простору, який зберігає семантичні властивості зображень і водночас є стійким до шумових спотворень. Центральним елементом регуляризації є дивергенція Кульбака–Лейблера, що примушує латентний розподіл наближатися до заданого апріорного. Така регуляризація впорядковує структуру ознак, забезпечує гладкий латентний простір, сприяє узагальненню моделі та зменшує ризик перенавчання. Без цього компонента варіаційний автокодувальник фактично перетворився б на звичайний автоенкодер, що не має чітко визначеної структури латентних репрезентацій і схильний до нестабільної реконструкції.

Важливим розширенням класичної регуляризації є  $\beta$ -VAE, де введення коефіцієнта  $\beta$  дозволяє контролювати силу впливу KL-дивергенції.

Збільшення цього параметра посилює структурованість латентного простору та сприяє *disentanglement*, хоча може погіршувати точність реконструкції.

Зменшення  $\beta$ , навпаки, підвищує якість відновлення за рахунок менш чітко структурованого латентного розподілу. У задачах денойзингу оптимальне значення  $\beta$  дає змогу знайти рівновагу між збереженням деталей і стійкістю моделі.

Ще одним дієвим прийомом є поступове збільшення регуляризації (KL-annealing). На початкових етапах навчання регуляризаційний тиск є значно слабшим, тому модель фокусується на якісній реконструкції. У подальшому вплив KL-штрафу зростає, і модель починає формувати впорядковану латентну структуру. Це допомагає уникнути явища «posterior collapse», коли енкодер перестає кодувати корисну інформацію.

Додатковими механізмами регуляризації виступають Dropout та додавання невеликого шуму до активацій чи ваг мережі. Dropout знижує ризик перенавчання, а noise injection привчає модель розпізнавати структуру у присутності флуктуацій, що позитивно впливає на здатність працювати з реальним шумом. Якщо латентний простір моделі поділено на семантичну та шумову частини, регуляризація також спрямована на те, щоб кожен підпростір відповідав саме своїй функції: семантика не

змішувалася з шумом, а шум кодувався окремо. Подібні підходи застосовуються в розширених варіантах VAE, таких як DVAE, Factorized VAE та  $\beta$ -TCVAE.

### Налаштування (tuning) параметрів

Параметри моделі суттєво впливають на стабільність навчання та якість реконструкції. Одним з найважливіших є розмір латентного простору. Якщо він занадто малий, модель втрачає здатність відображати важливі семантичні ознаки, що призводить до надмірно узагальнених, спрощених реконструкцій. Навпаки, надмірно великий латентний простір ускладнює регуляризацію, сприяє появі шумових компонентів у семантичній частині та підвищує ризик перенавчання. Оптимальне значення підбирають емпірично, за результатами тестування на метриках якості.

Архітектура моделі має суттєве значення. Глибокі згорткові мережі дозволяють краще відтворювати складні структури та текстури, однак надмірна глибина може спричинити нестабільність або перенавчання. Кількість каналів визначає можливість моделі відображати деталі, а використання residual-блоків або U-Net-подібних з'єднань допомагає зберігати просторову інформацію.

Важливим є вибір оптимізатора та параметрів навчання: Adam вважається найстабільнішим варіантом, проте результат значною мірою залежить від правильно підбраного learning rate. Надто великий крок може руйнувати латентний простір, тоді як надто малий — сповільнює збіжність.

У задачах відновлення зображень важливу роль відіграє також характер та інтенсивність шуму, що подається моделі. Різні типи шумів (гаусівський, імпульсний, спекл-шум, комбіновані варіанти) мають власні особливості та потребують різних стратегій навчання. Шум може додаватися безпосередньо до вхідних зображень, кодуватися окремим енкодером або вводитися в латентний простір. Вибір функцій активації також впливає на результат: ReLU та LeakyReLU добре зарекомендували себе у денойзингу, тоді як sigmoid використовується на виході декодера при нормованих інтенсивностях.

У розділених латентних моделях важливо правильно визначити пропорцію між розміром семантичного та шумового підпросторів, щоб уникнути взаємного змішування сигналу та спотворень.

### Loss-функції VAE

Функція втрат визначає, як саме модель навчається і які характеристики зображення вважаються важливими для збереження. Стандартна loss-функція складається з реконструкційної частини та KL-регуляризації. Реконструкційний термін вимірює різницю між оригінальним і відновленим зображеннями, тоді як KL-компонент забезпечує впорядкованість латентного простору. У задачах денойзингу такої комбінації часто недостатньо, тому loss-функцію розширюють додатковими компонентами.

Класичні реконструкційні втрати базуються на L2-помилці (MSE), яка добре пригнічує гаусівський шум, проте схильна до надмірного згладжування та появи «мільних» реконструкцій. Альтернативою є L1-помилка (MAE), яка краще зберігає краї та локальні структури. Комбінація L1 і L2 дозволяє поєднати їхні переваги.

Оскільки людське сприйняття фокусується не на окремих пікселях, а на структурі зображення, до функції втрат додають структурні метрики. Найпоширенішою є SSIM, яка враховує контраст, локальну структуру та кореляцію яскравостей. Її розширена версія MS-SSIM оцінює зображення на кількох масштабах, покращуючи відтворення текстур.

Для підвищення візуальної якості реконструкції широко використовуються перцептивні втрати. Вони порівнюють активації проміжних шарів попередньо навченої нейромережі (наприклад, VGG-16) і дозволяють моделі зберігати семантичні структури, навіть якщо піксельні відмінності залишаються. Це забезпечує природніший вигляд відновлених зображень.

У задачах денойзингу часто додається окремий штраф за шум, який змушує модель активніше прибирати спотворення. Якщо модель поєднує VAE та GAN, до функції втрат включається adversarial loss, що підсилює реалістичність результатів. Додатково може використовуватися Total Variation Loss, який згладжує різкі локальні переходи між пікселями.

Узагальнена loss-функція є багатокомпонентною та збалансованою таким чином, щоб модель одночасно зберігала структуру зображення, пригнічувала шум і підтримувала впорядкований латентний простір. Саме правильне поєднання цих компонентів є ключем до високої ефективності варіаційного автокодувальника в задачі відновлення зашумлених зображень відображається за формулою (2.1).

$$L = E[\|x - \hat{x}\|^2] + \alpha D_{KL}(q(z|x) || p(z)) + \beta E[\|y - x\|^2] + \gamma(1 - SSIM(x, \hat{x})) \quad (2.1)$$

Комбіновані та гібридні функції втрат демонструють найвищу ефективність, оскільки поєднують у собі кілька взаємодоповнювальних компонентів. Такий підхід дозволяє одночасно контролювати точність реконструкції, впорядкованість латентного простору, збереження структурних особливостей та ступінь пригнічення шуму.

У межах однієї функції втрат можуть поєднуватися реконструкційні терміни на основі L1 або L2-норми, регуляризаційний член у вигляді KL-дивергенції, складова, що враховує структурну подібність між зображеннями (SSIM), а також перцептивна частина, яка забезпечує збереження високорівневих ознак.

Додатково включається штраф за шум, що змушує модель активніше придушувати спотворення. Завдяки такому синтезу різних критеріїв модель отримує збалансований сигнал помилки й навчається відтворювати зображення не лише з мінімальною числовою похибкою, а й з високою візуальною якістю.

Зважування цих компонент здійснюється через коефіцієнти  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\lambda$  — кожен з яких можна налаштувати.

Ця форма функції втрат є однією з найефективніших у задачах відновлення зображень, оскільки водночас дозволяє моделі контролювати структурні особливості зображення, керувати рівнем шуму, формувати впорядкований і стабільний латентний простір та зберігати важливі візуальні деталі під час реконструкції.

## 2.4 Вибір інструментів, середовища програмування та бібліотек.

Розроблення методу оптимізації процесу відновлення зображень із шумом на основі архітектури варіаційного автокодувальника потребує ретельно підбраного програмного забезпечення, яке забезпечило б можливість ефективної реалізації складних обчислювальних процедур, гнучкої побудови моделей глибокого навчання та проведення широкого спектра експериментів. Вибір інструментів у цьому дослідженні був обумовлений вимогами до масштабованості, продуктивності, зручності розробки та наявності сучасних бібліотек, що підтримують операції на графічних процесорах. Особлива увага приділялася інструментам, які дозволяють легко реалізовувати складні нейромережеві архітектури, зокрема моделі з розділеним латентним простором, комбінованими функціями втрат та розширеними механізмами регуляризації.

Основною мовою програмування для реалізації моделі було обрано Python, оскільки ця мова фактично є індустріальним стандартом для задач машинного навчання, комп'ютерного зору та обробки цифрових даних. Python має надзвичайно широку екосистему бібліотек, які забезпечують підтримку для всіх етапів роботи з моделями: попереднього опрацювання даних, побудови нейронних мереж, виконання обчислень, оптимізації, збереження результатів та візуалізації. Серед ключових переваг Python — гнучкість синтаксису, можливість швидкої розробки прототипів, легка інтеграція зі сторонніми фреймворками та підтримка як центральними, так і графічними процесорами. Ці властивості роблять Python оптимальним вибором для реалізації варіаційного автокодувальника та експериментального дослідження параметрів моделі.

Для побудови, навчання та тестування варіаційного автокодувальника було обрано фреймворк PyTorch — одну з найпотужніших і найбільш поширених платформ для глибинного навчання. PyTorch забезпечує автоматичне диференціювання, що дозволяє реалізувати механізм зворотного поширення помилки без ручного визначення похідних. Крім того, фреймворк підтримує оптимізовані тензорні операції з можливістю виконання на GPU, що суттєво

прискорює процес навчання великих моделей. Архітектура PyTorch є модульною та об'єктно-орієнтованою, що дає змогу легко створювати власні класи енкодера, декодера, репараметризаційних модулів та латентних підпросторів, адаптувати функції втрат і реалізовувати навіть нетривіальні варіації VAE. Особливе значення мала можливість змінювати структуру латентного простору, розширювати модель додатковими компонентами та експериментувати з різними гіперпараметрами без суттєвої модифікації вихідного коду.

Важливим елементом інфраструктури, що супроводжує роботу моделі, є бібліотеки для обробки даних. На етапі формування набору зображень і підготовки їх до тренування активно використовувалися NumPy та Pandas, які забезпечують ефективну обробку багатовимірних масивів та табличних структур. Завдяки ним стало можливим виконувати статистичний аналіз даних, формувати вибірки, нормалізувати зображення та виконувати чисельні операції з високою точністю.

Для спеціалізованих операцій із зображеннями були використані бібліотеки torchvision та scikit-image. Вони забезпечують засоби для масштабування, трансформації, конвертації, а також генерування штучних шумів, необхідних для навчання моделі денойзингу. Оскільки варіаційний автокодувальник працює не лише з чистими зображеннями, але й зі спотвореними версіями, саме ці бібліотеки надали можливість реалізувати різні сценарії зашумлення: від додавання гаусівського або пуассонівського шуму до моделювання імпульсних або мультиплікативних спотворень.

Візуалізація відіграє ключову роль у дослідженні моделей глибокого навчання. Тому у роботі використовувалися Matplotlib і Seaborn, які надали змогу відображати графіки зміни функції втрат, середньоквадратичних похибок, показників структурної подібності та інших метрик. Візуальні методи аналізу дозволили оцінити збіжність моделі, стабільність навчання, вплив регуляризації та параметрів оптимізатора. Крім того, за їх допомогою здійснювалося порівняння реконструкцій, отриманих від різних конфігурацій архітектури VAE, та оцінювання рівня втрати деталей або нівелювання шуму на різних етапах тренування.

Процес розробки, налагодження та експериментального тестування був організований у середовищах JupyterLab та Google Colab. Ці інструменти забезпечили інтерактивне виконання коду, що виявилось особливо корисним під час роботи з багатоступневими обчисленнями і складними архітектурами. JupyterLab дозволяв покроково виводити результати, фіксувати проміжні метрики та швидко коригувати гіперпараметри, а Google Colab надавав доступ до графічних процесорів, що значно пришвидшило тренування моделей. Використання GPU було критично важливим, адже навчання варіаційних автокодувальників на великих наборах зображень може бути дуже ресурсозатратним. Завдяки обчисленням на графічних процесорах вдалося скоротити час тренування в декілька разів і отримати можливість проводити достатньо тривалі експерименти.

Під час реалізації методу важливою є можливість зберігати параметри моделі, серіалізувати ваги, логувати результати та забезпечувати відтворюваність експериментів.

Ці функції реалізовані у PyTorch через модулі `torch.save`, `torch.load`, а також сторонні інструменти `logging` та `tqdm` для відстеження прогресу. Комбінація цих засобів дозволила структурувати процес навчання і зберігати всі проміжні результати без ризику їх втрати.

Вибраний технологічний стек дав змогу реалізувати складну архітектуру варіаційного автокодувальника з розділеним латентним простором, комбінованими функціями втрат та різними конфігураціями глибини мережі. Завдяки підтримці GPU вдалося навчати моделі, які містять значну кількість параметрів, а засоби візуалізації та логування забезпечили глибокий аналіз отриманих результатів.

Також використані інструменти дозволили легко модифікувати структуру моделі, експериментувати з гіперпараметрами, змінювати типи шуму та тестувати різні архітектурні рішення. Усе це разом забезпечило можливість ефективно реалізувати оптимізацію процесу відновлення зображень та виконати повноцінний цикл наукового експерименту — від формування датасета до аналізу результатів.

## 2.5. Формування наборів даних та методи їх штучного зашумлення.

Для ефективного навчання варіаційного автокодувальника необхідно сформувати репрезентативний та структурно багатий набір даних, який відображає широкий спектр варіацій зображень та різних типів шумів, характерних для реальних умов отримання цифрової інформації. Якість і різноманітність вихідного датасета суттєво впливають на здатність моделі до узагальнення, стабільність латентного простору та точність реконструкції. Крім того, для задачі денойзингу критично важливо створити коректні пари «чисте — зашумлене» зображення, які дадуть змогу моделі навчитися відокремлювати структурну інформацію від шумових флуктуацій.

Формування якісної вибірки передбачає відбір такого набору зображень, який би відображав достатню різноманітність текстур, контурів, кольорових відтінків та складності сцен. У залежності від поставленого завдання можуть використовуватися загальнодоступні набори даних, такі як CIFAR-10, STL-10, MNIST, ImageNet Subset, або власні тематичні колекції, отримані з відкритих джерел чи сформовані шляхом самостійного збору даних. Вибір набору даних визначається також тим, наскільки складні структури модель повинна вміти відновлювати. Наприклад, CIFAR-10 містить відносно прості зображення низької роздільності, тоді як ImageNet або спеціалізовані доменні датасети (медичні, супутникові, промислові) дають змогу моделі тренуватися на складних, багатошарових структурах, що підвищує її потенціал у реальних застосуваннях.

Перед подачею до моделі всі зображення проходять стандартизований процес попередньої обробки, що забезпечує однорідність даних та стабільність тренування. До основних етапів препроцесингу належать:

- масштабування розмірів, що дає змогу перетворити всі зображення до єдиного формату та спростити обробку;
- нормалізація інтенсивностей пікселів, зазвичай у діапазон  $[0, 1]$  або  $[-1, 1]$ , що прискорює збіжність моделі;

- конвертація до grayscale або залишення RGB-каналів, залежно від того, чи є модель чутливою до колірних ознак;
- аугментація, яка включає легкі випадкові зміни, такі як обертання, горизонтальні віддзеркалення, зміни яскравості та контрасту, невеликі афінні перетворення тощо.

Аугментація сприяє підвищенню стійкості моделі до варіацій сцени та допомагає уникати перенавчання, оскільки збільшує ефективний розмір навчального набору. У дослідженнях з глибинного навчання показано, що правильно підібрана аугментація значно покращує результати реконструкції і робить модель більш робастною щодо реальних спотворень.

Після препроцесингу набір даних зазвичай ділиться на три частини: тренувальну, валідаційну та тестову. Такий розподіл дозволяє адекватно оцінювати здатність моделі узагальнювати інформацію, контролювати перенавчання та забезпечує незалежний набір для фінальної оцінки якості.

Варіаційний автокодувальник навчається на основі пар, де  $x$  — вихідне чисте зображення, а  $y$  — його зашумлена версія. Саме через цю пару модель дізнається, яку інформацію необхідно відновити, а яку — усунути. Оскільки в реальних умовах шум не завжди є доступним для прямого вимірювання, його моделюють штучно.

Штучне зашумлення дозволяє не лише контролювати інтенсивність та тип шуму, але й розширити простір можливих спотворень, що збільшує стійкість моделі. Зокрема, у роботі розглядаються такі типи шумів:

Гаусівський шум є одним із найпоширеніших видів випадкових спотворень і виникає внаслідок флуктуацій у сенсорах зображення або передачі сигналу. Він моделюється як додавання до кожного пікселя випадкової величини із нормальним розподілом.

Гаусівський шум широко використовується в дослідженнях денойзингу завдяки своїй простоті та універсальності. Однак, його усунення потребує від моделі здатності зберігати структуру контурів, оскільки надмірне згладжування може призвести до втрати важливих деталей.

Цей тип шуму моделює раптові збої сенсорів або втрату окремих пікселів. На зображенні він проявляється як поодинокі білі та чорні точки, що робить його складнішим для відновлення, ніж гаусівський шум. Імпульсний шум є надзвичайно важливим у тестуванні денойзингових моделей, оскільки вимагає від VAE здатності локально відновлювати втрачені дані без деформації основної структури.

Спекл-шум має мультиплікативну природу та часто зустрічається у медичних даних (наприклад, ультразвукових зображеннях) або радарних супутникових знімках. Його характер складніший, ніж у попередніх типів шумів, тому моделювання спекл-шуму суттєво підвищує рівень узагальнення моделі та робить її стійкою до високочастотних флуктуацій.

Пуассонівський або фотонний шум залежить від інтенсивності сигналу й виникає за умови низького освітлення. Його моделювання дозволяє імітувати спотворення, характерні для камер зі слабким світлочутливим сенсором, а також систем відеоспостереження або астрономічної зйомки.

Для підвищення стійкості моделі використовують кілька рівнів інтенсивності шуму: слабкий, середній та сильний. Такий підхід дозволяє моделі адаптуватися до різних сценаріїв і виконувати реконструкцію навіть у складних умовах. Практика показує, що одночасне використання кількох рівнів шуму суттєво зменшує «вразливість» VAE до спотворень, які відрізняються від тренувальних.

У сучасних фреймворках, таких як PyTorch, штучне зашумлення реалізується безпосередньо під час формування батчів, що дозволяє динамічно генерувати нові варіанти спотворень. Таким чином, кожне чисте зображення може отримати кілька унікальних зашумлених копій протягом тренування, що підвищує різноманітність даних і зменшує ризик перенавчання.

Після здійснення всіх етапів preprocessing та зашумлення формуються пари  $(x, y)$ , де  $x$  — чисте зображення, а  $y$  — його зашумлена версія. Така структура даних є базовою для навчання варіаційного автокодувальника, оскільки через реконструкційну функцію втрач моделі навчається відокремлювати зміст зображення.

Комплексність штучно сформованого набору, включно з варіативністю типів шуму, рівнями їх інтенсивності, різноманіттям контенту та структурною складністю сцен, забезпечує високу узагальнювальну здатність моделі та її адаптивність до реальних даних.

## 3 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

### 3.1. Опис процесу навчання моделі.

Процес навчання варіаційного автокодувальника, ґрунтується на поетапному формуванні інформативного латентного простору та адаптації параметрів моделі до властивостей вхідних даних. Навчання моделі передбачає цілеспрямоване відокремлення семантичної інформації від шумових компонентів, а також поступову оптимізацію архітектури через реконструкцію зображень та регуляризацію латентного простору. Такий підхід дозволяє підвищити якість відновлення, збільшити стійкість моделі до різних рівнів зашумлення та забезпечити узагальнюваність результатів.

На першому етапі формується навчальна вибірка. Вона містить пари зображень: оригінальні («чисті») та їх зашумлені версії. Створення штучно зашумлених прикладів є важливою частиною підготовки даних, оскільки дозволяє моделі зустріти широкий спектр можливих шумових патернів. Для кожного зображення генеруються декілька варіацій із різним рівнем та характером шуму: від легких флуктуацій до інтенсивних спотворень. Це дає змогу VAE не прив'язуватися до конкретної інтенсивності шуму і формувати гнучку здатність до відновлення. На цьому ж етапі здійснюється попереднє опрацювання даних: нормалізація значень пікселів, приведення до стандартного розміру, масштабування інтенсивностей та, за потреби, аугментація. Сюди може входити обертання, віддзеркалення, варіації контрасту чи яскравості, що збільшує різноманіття навчального набору та покращує узагальнювальні властивості моделі.

Після підготовки даних модель переходить до основного етапу навчання. Енкодер приймає зашумлене зображення  $y$  і перетворює його на параметричне латентне представлення. На відміну від класичних автоенкодерів, де латентне представлення є фіксованим вектором, варіаційний автокодувальник формує два параметри для кожної компоненти латентного простору: середнє значення та

дисперсію. Такий підхід дозволяє моделі враховувати невизначеність та варіативність даних. У роботі використано модифікований латентний простір, поділений на дві частини:  $z_{\text{semantic}}$ , який відповідає за зміст і структуру зображення, та  $z_{\text{noise}}$ , який моделює шумові компоненти. Така декомпозиція дозволяє чітко розмежувати корисну та небажану інформацію.

Перед передачею латентного коду декодеру здійснюється репараметризація. Вона необхідна через стохастичну природу латентного простору VAE: замість безпосереднього вибіркового генератора випадкових чисел використовується диференційовний прийом, що дозволяє обчислювати градієнти та проводити навчання методом зворотного поширення помилки. Результат репараметризації — латентна змінна  $z$ , яка містить у собі структуровану модель вмісту та шуму, отриману відповідно до статистичних параметрів, обчислених енкдером.

Декодер отримує латентну змінну  $z$  і формує на її основі реконструйоване зображення. Саме на цьому етапі відбувається основний процес відновлення: модель намагається перетворити латентне представлення на якомога чистіший та візуально якісний результат. Важливо, що декодер працює не над безпосереднім очищенням шуму, а над відтворенням структури зображення на основі інформації, що міститься у латентному просторі. Це забезпечує стійкість моделі до випадкових шумів та покращує якість узагальнення на нові дані.

Оцінювання якості реконструкції проводиться за допомогою функції втрат, яка включає кілька компонентів. Реконструкційна складова визначає, наскільки точно декодер відтворює оригінальний образ, порівнюючи реконструкцію з чистим зображенням.

Регуляризаційна частина у вигляді KL-дивергенції забезпечує структуру латентного простору та запобігає перенавчанню, створюючи гладку та впорядковану латентну репрезентацію.

Додаткові компоненти функції втрат включають структурну складову, таку як SSIM-втрата, яка дозволяє зберігати контури та геометрію об'єктів, а також штрафи, що відповідають за пригнічення шумових компонентів. Особливо важливою є та

частина функції втрат, яка контролює збереження відмінностей між  $z_{\text{semantic}}$  і  $z_{\text{noise}}$ , запобігаючи змішуванню інформації про зміст та шум.

Процес оптимізації виконується через градієнтний спуск з використанням оптимізатора Adam (див.Рис 3.1), який завдяки адаптивній зміні моментів градієнтів забезпечує стабільну та швидку збіжність. Правильний вибір швидкості навчання є критично важливим: надто велика швидкість може спричинити нестабільність і «розвал» латентного простору, тоді як надто мала — призводить до повільного, неефективного навчання.

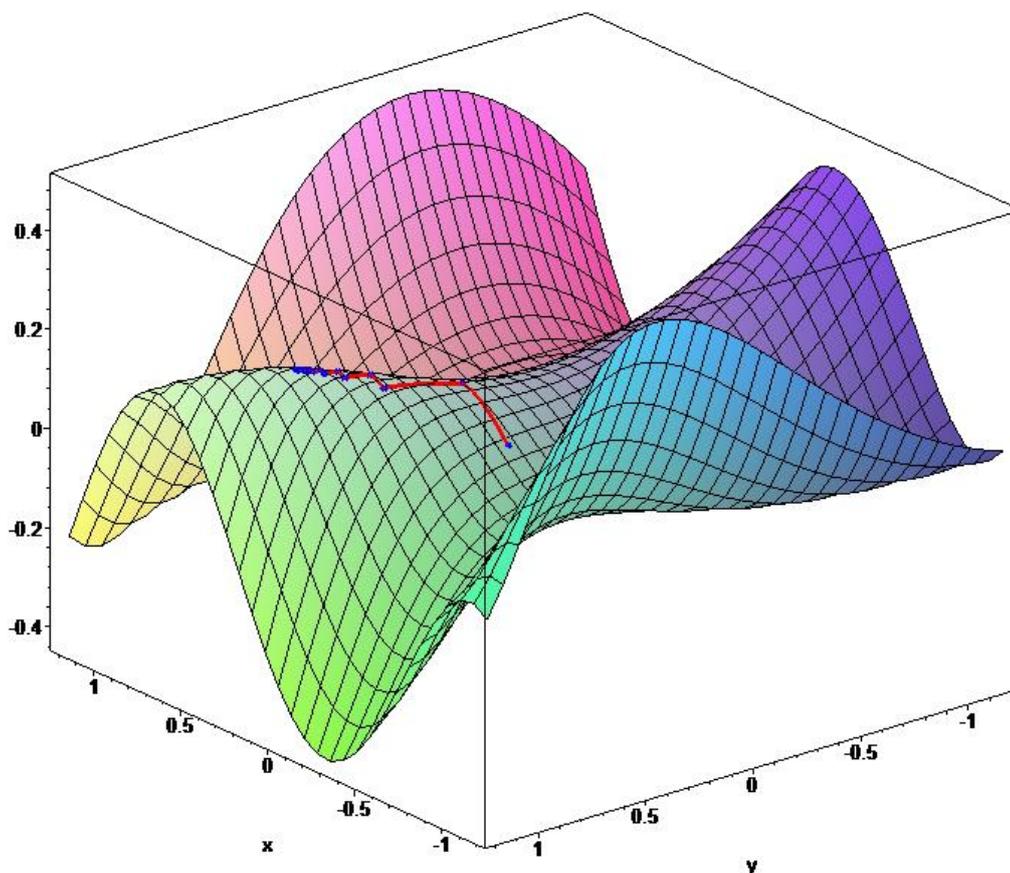


Рис. 3.1 Градієнтний спуск

Для підтримання стабільного процесу використовуються техніки регуляризації навчання, зокрема KL-annealing, який поступово збільшує вплив регуляризаційного терміна. Це дозволяє моделі на початкових етапах зосередитися на реконструкції, а згодом формувати впорядкований латентний простір.

Особливу увагу в тренувальному процесі приділено контролю незмішування семантичних та шумових компонентів. До функції втрат включено додаткові штрафи, які забезпечують мінімальну кореляцію між цими просторами, що дозволяє моделі точніше відокремлювати значущі ознаки зображення від випадкових спотворень. Це важливо для запобігання ситуаціям, коли шумові патерни «потрапляють» у семантичний простір і порушують якість реконструкції.

Під час навчання якість роботи моделі відстежується на валідаційній вибірці. Це дає змогу оцінити, наскільки добре модель узагальнює інформацію, а також вчасно помічати перенавчання. Важливим доповненням до числової оцінки є візуальний аналіз реконструкцій, який дозволяє виявляти артефакти, розмиття або інші недоліки, які не завжди відображаються в числових метриках.

Після досягнення стабільного рівня якості модель зберігається разом зі своєю архітектурою та вагами. Це дозволяє використовувати її для реконструкції нових зображень, а також інтегрувати в прикладні системи, які потребують автоматичного очищення даних від шуму. Завдяки гнучкості архітектури та добре структурованому латентному простору модель здатна працювати з широким спектром шумових патернів і забезпечувати високу якість відновлення в реальних умовах.

### **3.2. Порівняння різних конфігурацій архітектури.**

У межах даного дослідження було здійснено комплексне порівняння кількох архітектурних конфігурацій варіаційного автокодувальника з метою визначення оптимальної структури моделі, здатної забезпечити найвищу якість відновлення зображень, спотворених шумом. Оскільки різні параметри мережі по-різному впливають на здатність моделі до реконструкції, було досліджено цілу низку змінних: глибину енкодера та декодера, кількість згорткових блоків, застосування залишкових (residual) модулів, розмір латентного простору, його структуру, інтенсивність регуляризації та механізми передачі просторової інформації на зразок skip-з'єднань. Об'єктив порівняння полягав не лише у визначенні найточнішої архітектури, але й у виявленні компромісів між якістю реконструкції, стійкістю до

різних типів шуму, обчислювальною ефективністю та узагальнюваною здатністю моделі.

Першим кроком у процесі порівняння стала оцінка базової архітектури VAE, побудованої на класичному підході з одним енкодером, стандартним латентним простором фіксованого розміру та декодером дзеркальної структури. Такий варіант дозволив отримати фундаментальні показники, які надалі виступали відправною точкою для порівняння з іншими, складнішими модифікаціями. Аналіз показав, що базова архітектура ефективно пригнічує низькоінтенсивний шум, однак її можливості виявилися недостатніми для роботи зі структурованими або високочастотними шумовими патернами.

Латентний простір стандартного VAE не має внутрішнього розділення на семантичні та шумові компоненти, унаслідок чого відбувалося змішування текстур, контурів та шумових змін. Це призводило до появи типових артефактів: розмиття, втрати дрібних деталей, надмірної згладженості або, навпаки, «збереження» частини шуму у відновленому зображенні. Незважаючи на певний рівень реконструкційної точності, базова архітектура проявила обмежену здатність відокремлювати корисний сигнал від спотворень.

Другою групою досліджуваних конфігурацій стали варіанти зі збільшеною глибиною мережі. Сюди входили архітектури, в яких енкодер і декодер містили більшу кількість згорткових блоків та проміжних шарів. Додавання нових рівнів дало змогу моделі виділяти більш високорівневі ознаки та краще вловлювати дрібні структурні елементи зображення. Такі мережі демонстрували значно точнішу реконструкцію текстур, покращене відтворення контурів та меншу кількість артефактів зображеннях.

Однак зростання глибини має і зворотний бік: збільшена кількість параметрів зробила модель більш чутливою до перенавчання. Без належної регуляризації поглиблені архітектури починали «запам'ятовувати» шумові патерни тренувальної вибірки, що зменшувало їх узагальнювальну здатність. Зокрема, при роботі з новими типами шуму такі моделі демонстрували погіршення якості реконструкцій.

Таким чином, хоча підвищена глибина покращувала точність відновлення, вона не була оптимальною в аспекті продуктивності та стійкості моделі.

Найбільш перспективним напрямом модифікації виявився підхід, у якому латентний простір був поділений на семантичну та шумову компоненти. У цій конфігурації зображення у процесі кодування перетворювалося не на один латентний вектор, а на два окремих підпростори: один моделював зміст ( $z_{\text{semantic}}$ ), а інший — шум ( $z_{\text{noise}}$ ).

Така декомпозиція дозволила досягти концептуально нового рівня відновлення.

Результати показали, що модель зі структурованим латентним простором краще ідентифікувала шумові характеристики та більш точно зберігала структуру зображення. Семантична частина латентного коду зберігала інформацію про форми, контури, текстури, а шумова частина акумулювала випадкові спотворення, які модель могла ідентифікувати та пригнічувати під час реконструкції. Цей підхід дозволив моделі не лише відтворювати зображення, близькі до оригіналу, але й робити це стабільно у випадках, коли тип шуму помітно відрізнявся.

Крім того, саме ця конфігурація продемонструвала найменше змішування семантичних та шумових ознак. Використання спеціальних регуляризаційних термінів, що контролювали незалежність двох підпросторів, забезпечило чітке розмежування інформації, що суттєво зменшило появу спотворень та артефактів у реконструкції.

Окремим етапом дослідження стало оцінювання впливу розміру латентного простору та співвідношення між його семантичною та шумовою частинами. Експерименти виявили, що занадто малий семантичний простір призводив до того, що модель не могла точно відтворювати важливі деталі зображення. Особливо помітною була втрата текстур, контурів або дрібних об'єктів. З іншого боку, надмірно великий шумовий підпростір зменшував ефективність пригнічення шуму, оскільки модель починала зберігати у ньому не лише спотворення, але й частину корисної інформації.

Оптимальне співвідношення розмірів двох підпросторів дозволило досягти балансу між точністю реконструкції та стійкістю до шуму. Виявлено, що збільшення семантичного компонента дає змогу краще передавати структуру, а помірний шумовий компонент — ефективніше моделювати флуктуації сигналу. Такий підхід зменшив втрату інформації та забезпечив стійкі результати за різних рівнів зашумлення.

Додатковим напрямом оптимізації стало тестування архітектур із включенням skip-з'єднань та residual-блоків. Skip-з'єднання надавали можливість напряму передавати високороздільні просторові ознаки з ранніх шарів енкодера до відповідних рівнів декодера. Це суттєво покращувало відтворення локальних структур, підвищувало чіткість контурів та зменшувало розмиття, характерне для класичних VAE.

Residual-блоки, своєю чергою, сприяли стабілізації навчання, допомагали уникати проблеми затухання градієнтів та дозволяли ефективно використовувати глибші конфігурації без надмірного ризику перенавчання. Комбінація цих механізмів продемонструвала значне покращення візуальної якості реконструкцій, особливо в умовах складного або неоднорідного шуму.

Таблиця 3.1

## Порівняння конфігурацій методів

| Конфігурація архітектури                  | Опис моделі  | Переваги   | Недоліки   | Оцінка ефективності               |
|---|--|--|--|-----------------------------------|
| Латентний простір                         | Енкодер і декодер середньої глибини, один латентний простір без поділу на семантику та шум           | Простота реалізації, швидке навчання, низькі обчислювальні витрати   | Змішування семантичних і шумових компонент; втрата деталей; нечіткі контури; слабка стійкість до складного шуму  | Низька                            |
| Поглиблений VAE                           | Глибша CNN-архітектура з більшою кількістю фільтрів і резидуальних блоків                            | Краще моделює структуру та підвищена точність реконструкції; менше розмиття  | Висока чутливість до перенавчання; значно більші обчислювальні витрати; нестабільність без сильної регуляризації | Середня–висока, залежить від шуму |
| VAE зі структурованим латентним простором | Два незалежні підпростори для семантики та шуму; окремі енкодери або окремі гілки всередині енкодера | Чітке розділення сигналу та шуму; висока точність реконструкції; збереження деталей; стійкість до різних типів шуму; | Підвищена складність реалізації; більше параметрів; потребує ретельного налаштування функції втрат               | Висока                            |



Таблиця 3.1

|                       |  |  |  |                |
|-----------------------|--|--|--|----------------|
| VAE з skip-конекціями | Декодер отримує додаткову інформацію з енкодера      | Найкраще зберігає просторові залежності                              | Більший обсяг пам'яті та обчислень                         | Висока         |
| VAE-Perceptual Loss   | Додано компонент верцептивної втрати                 | Реалістичні реконструкції збереження високорівневих ознак та текстур | Більший час тренування; Залежність від перенавчання моделі | Середня-висока |
| $\beta$ -VAE          | Посилена регуляризація збільшення параметрів $\beta$ | Краща стабільність та впорядкований латентний простір                | Зниження якості реконструкції при втраті деталей           | Середня        |

### 3.3. Аналіз впливу параметрів VAE на якість реконструкції.

Якість реконструкції даних у варіаційному автокодувальнику значною мірою залежить від вибору та налаштування його параметрів, які визначають структуру латентного простору, поведінку моделі під час навчання та здатність коректно відокремлювати шум від корисної інформації.

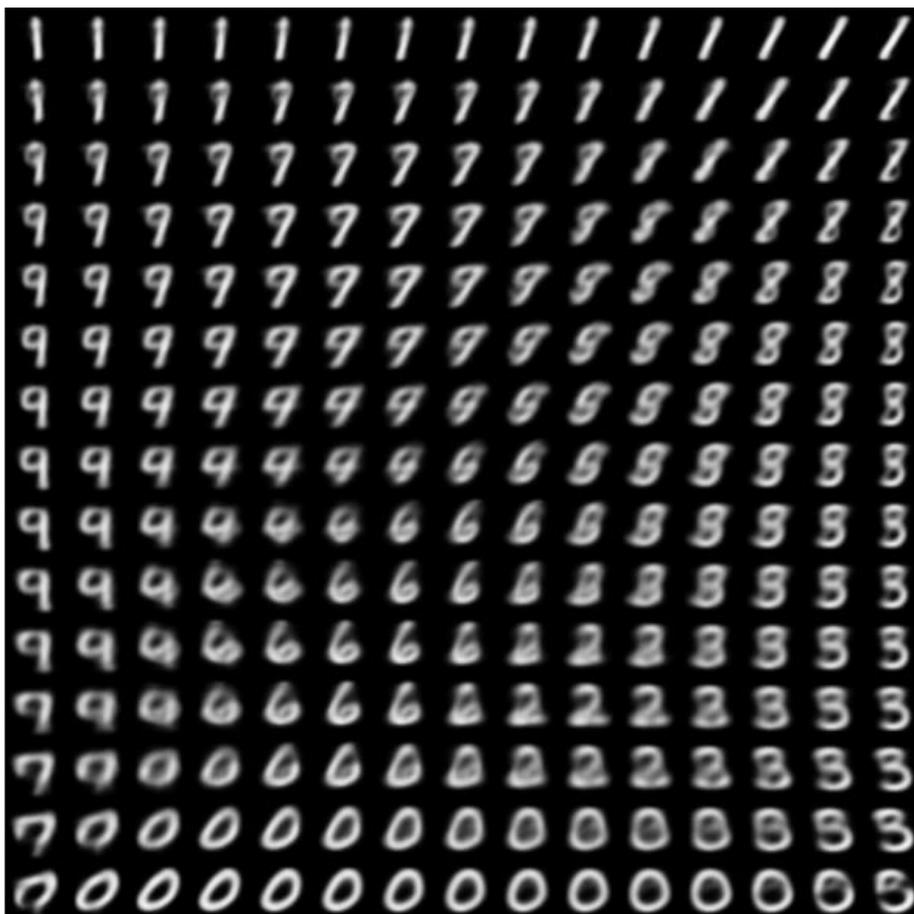


Рис. 3.2 Реконструкція зображення

Основними факторами, що впливають на кінцевий результат, є розмір латентного простору, характеристики енкодера та декодера, параметри регуляризації, вибір оптимізатора й швидкості навчання, а також інтенсивність та тип шуму, який використовується під час навчання.

Одним із ключових параметрів є розмір латентного простору, оскільки він визначає кількість інформації, яку модель може компактно закодувати. Занадто малий латентний вектор призводить до втрати високорівневих і дрібних структур,

внаслідок чого реконструкція стає надмірно згладженою. Якщо ж латентний простір є надто великим, модель починає зберігати занадто багато деталей, включно з елементами шуму, що ускладнює регуляризацію і погіршує якість очищення зображення.

Оптимальний розмір латентного представлення забезпечує баланс між здатністю передавати семантику та можливістю пригнічувати випадкові спотворення.

Важливим фактором є архітектурні параметри енкодера і декодера.

Глибші та ширші мережі здатні краще моделювати складні структури та текстури, що позитивно впливає на точність реконструкції. Водночас вони підвищують ризик перенавчання, особливо у випадках, коли вхідні дані містять різнотипний шум або мають низьку варіативність.

Використання резидуальних блоків чи механізмів пропускних зв'язків може покращити збереження просторових залежностей, проте потребує ретельної регуляризації, аби уникнути перенесення шуму через ці з'єднання.

Регуляризаційні параметри, зокрема сила KL-дивергенції або коефіцієнт  $\beta$  у  $\beta$ -VAE, суттєво впливають на структуру латентного простору.

Посилена регуляризація робить латентний простір більш впорядкованим і придатним для денойзингу, але водночас може негативно позначитися на можливості моделі передавати дрібні візуальні деталі.

Занадто слабка регуляризація, навпаки, покращує реконструкцію візуально, але робить модель менш стійкою до шуму та призводить до його часткової передачі у латентне представлення. Оптимальний вибір сили регуляризації є ключовим для формування структурованого латентного простору, у якому семантичні та шумові компоненти чітко розділені.

Важливими є параметри навчання — швидкість навчання, розмір пакета та тип оптимізатора.

Надто велика швидкість навчання може порушити стабільність латентного простору та призвести до коливань у реконструкції.

Занадто мала — сповільнює навчання і може зафіксувати модель у локальних мінімумах. Оптимізатор Adam зазвичай забезпечує кращу збіжність, проте також вимагає коректного налаштування моментів і швидкості навчання.

Розмір пакета впливає на стабільність градієнтів: великі пакети забезпечують більш плавне оновлення параметрів, але можуть призводити до переузгодженості з конкретними статистичними властивостями шуму.

Додатково на якість реконструкції впливають параметри, пов'язані з шумом: тип, рівень та спосіб подачі моделі.

Гаусівський шум легше піддається пригніченню за допомогою моделей, що оптимізуються під MSE, тоді як імпульсний або спекл-шум потребують глибших моделей або структурних компонентів у функції втрат. Якщо рівень шуму під час навчання занадто низький, модель не навчиться працювати із сильними спотвореннями; якщо ж занадто високий — вона зосередиться на відтворенні загальних контурів і втратить здатність відновлювати деталі.

У моделях зі структурованим латентним простором, де семантична і шумова складові розділені на окремі підпростори, важливо вибрати коректні розміри кожної частини. Надмірно великий шумовий підпростір дозволяє моделі кодувати надлишкову кількість флуктуацій, тоді як занадто маленький семантичний підпростір не може відобразити всю структуру зображення. Аналіз показує, що правильне співвідношення між розмірами підпросторів суттєво покращує здатність моделі пригнічувати шум та зберігати візуальні деталі.

Узагальнюючи, можна зазначити, що вплив параметрів VAE на якість реконструкції є комплексним і взаємозалежним. Жоден параметр не визначає результат самостійно — високу якість забезпечує лише збалансована конфігурація, у якій архітектурні особливості, регуляризаційні механізми та параметри навчання узгоджені між собою. Тільки за таких умов варіаційний автокодувальник здатний ефективно відновлювати дані, спотворені шумом, і демонструвати стабільність та високу точність реконструкції за різних умов використання.

### 3.4. Порівняння VAE з іншими методами відновлення даних.

Варіаційні автокодувальники є лише одним із класів моделей, що застосовуються для задач відновлення зашумлених зображень, і їх ефективність стає більш очевидною при порівнянні з іншими підходами — денойзинговими автоенкодерами (DAE) та згортковими нейронними мережами (CNN)(таблиця 1.1).

Денойзинговий автоенкодер є однією з найраніших моделей для реконструкції зображень, що працює за принципом прямого перетворення від зашумленого сигналу до його очищеної версії. Оскільки DAE не включає стохастичного латентного простору та не використовує регуляризацію у вигляді дивергенції Кульбака–Лейблера, модель не формує впорядкованих прихованих ознак та є менш стійкою до нових типів шуму. Вона добре працює з тренувальним шумом, але демонструє слабке узагальнення при зміні структури або інтенсивності спотворень.

Класичні CNN-підходи до денойзингу, зокрема архітектури U-Net та глибокі резидуальні мережі, забезпечують високу точність реконструкції завдяки глибоким згортковим блокам та skip-з'єднанням, які зберігають просторові залежності. Проте ці моделі є детермінованими та не моделюють розподільчу природу даних, що призводить до перенавчання на конкретних зразках шуму. Хоча CNN часто перевершують VAE за кількісними метриками реконструкції, вони значно гірше працюють у ситуаціях складної або невизначеної структури шуму.

На цьому тлі варіаційний автокодувальник вирізняється здатністю формувати стохастичний латентний простір, який задається параметризованим розподілом та контролюється KL-регуляризацією. Це дозволяє моделі відокремлювати стійкі семантичні ознаки від випадкових шумових компонентів і формувати впорядковані латентні представлення, що покращує узагальнення та стабільність реконструкції навіть на даних з новими типами спотворень.

У контексті задач денойзингу структурований латентний простір VAE допомагає моделі пригнічувати шум без відтворення випадкових флуктуацій, що є характерною проблемою DAE.

Крім того, VAE є генеративною моделлю, здатною виконувати семантичну інтерполяцію та моделювати невизначеність у вхідних даних, на відміну від класичних CNN та DAE, які працюють лише як детерміновані перетворювачі.

Це робить VAE більш гнучким інструментом для задач, де необхідно працювати з непередбачуваними або складними шумовими структурами.

Водночас важливо зазначити, що CNN-базовані моделі забезпечують кращу реконструкцію дрібних деталей, особливо якщо шум однорідний і добре представлений у тренувальній вибірці. DAE є простішими й швидшими, але суттєво програють у здатності до узагальнення.

Загалом аналіз показує, що VAE займає збалансовану позицію між високоточною, але менш гнучкою CNN-реконструкцією та простою, але нестійкою DAE-парадигмою.

Завдяки структурованому латентному простору та регуляризації, VAE забезпечує високу стабільність, здатність до генеративного моделювання та хорошу якість денойзингу за широким спектром шумів.

### **3.5. Оцінювання ефективності моделі за метриками.**

Оцінювання якості відновлення зображень є ключовим етапом аналізу роботи варіаційного автокодувальника, оскільки саме за допомогою метрик можна визначити, наскільки точно модель реконструює структуру зображення після усунення шуму.

Використання різних типів метрик забезпечує всебічне розуміння ефективності моделі, оскільки кожен показник відображає окремий аспект відновлення. Серед базових метрик, що традиційно застосовуються у задачах знешумлення, найбільш поширеними є середньоквадратична похибка (MSE), пікове співвідношення сигналу до шуму (PSNR) та структурна подібність (SSIM).

Середньоквадратична похибка (MSE) є класичним показником, який оцінює різницю між оригінальним та реконструйованим зображенням шляхом усереднення квадратів відхилень значень відповідних пікселів. Зниження MSE свідчить про

меншу числову різницю між зображеннями, однак ця метрика погано узгоджується з людським сприйняттям і не враховує локальних структурних особливостей. У багатьох випадках модель може демонструвати низьке MSE, але при цьому відтворювати розмиті зображення, оскільки метрика не реагує на втрату текстури або різкості.

На відміну від MSE, пікове співвідношення сигналу до шуму PSNR (див.рис 3.3) дає інтегральну оцінку рівня спотворень, що залишились після відновлення. Вища величина PSNR означає кращу якість зображення, оскільки зменшується інтенсивність шуму, присутнього у реконструкції. Ця метрика є особливо корисною для технічних порівнянь алгоритмів і широко застосовується в обробці сигналів та зображень. Проте, як і MSE, PSNR залишається малочутливим до локальних структурних змін і може неадекватно відображати якість відновлення візуально складних сцен.

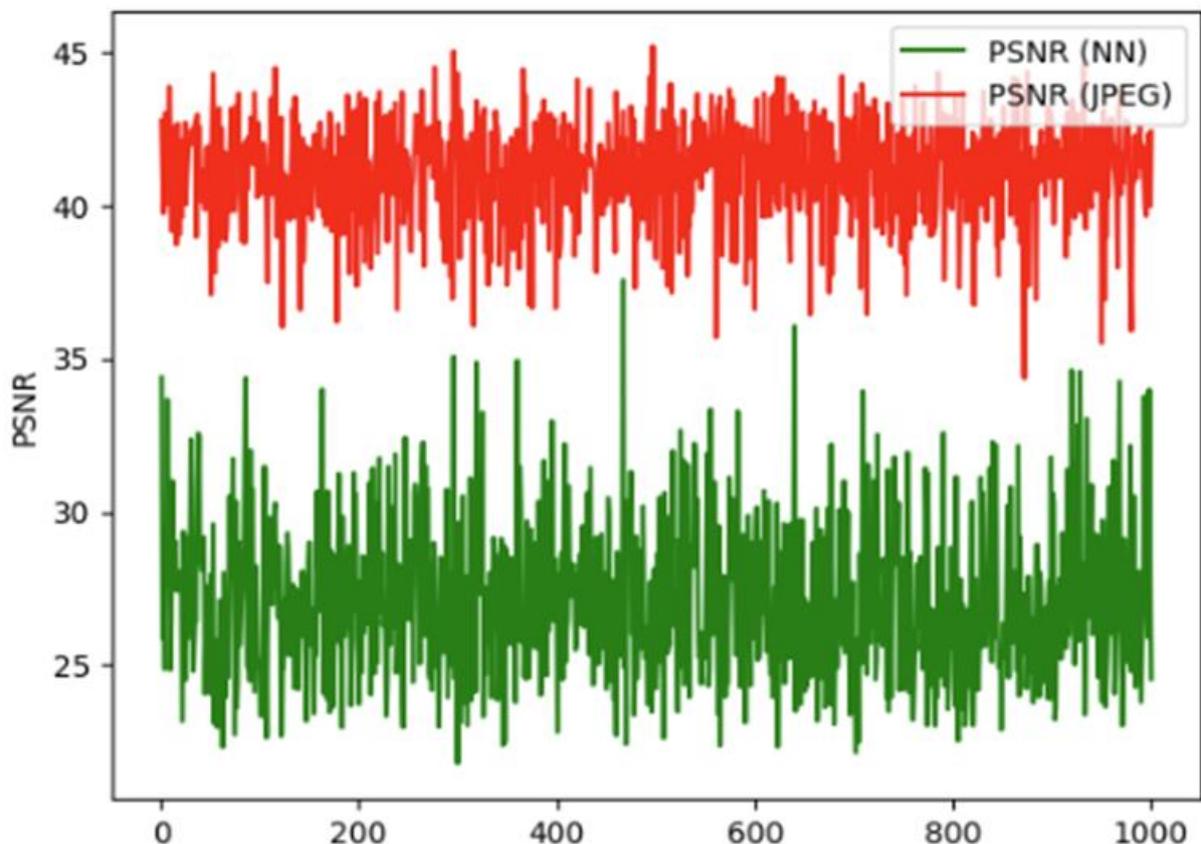


Рис. 3.3 Оцінка якості за метрикою PSNR

Найбільш інформативною у контексті зображень є метрика структурної подібності SSIM (див .рис. 3.4), яка порівнює оригінальне та реконструйоване зображення за трьома ключовими характеристиками: яскравістю, контрастом та локальною структурою. SSIM краще узгоджується з особливостями людського сприйняття, оскільки зосереджується на відтворенні структурних елементів, таких як контури, геометричні фрагменти та текстура. Високе значення SSIM свідчить про те, що модель зберігає семантичний зміст і локальні деталі навіть за наявності помірних числових відхилень.

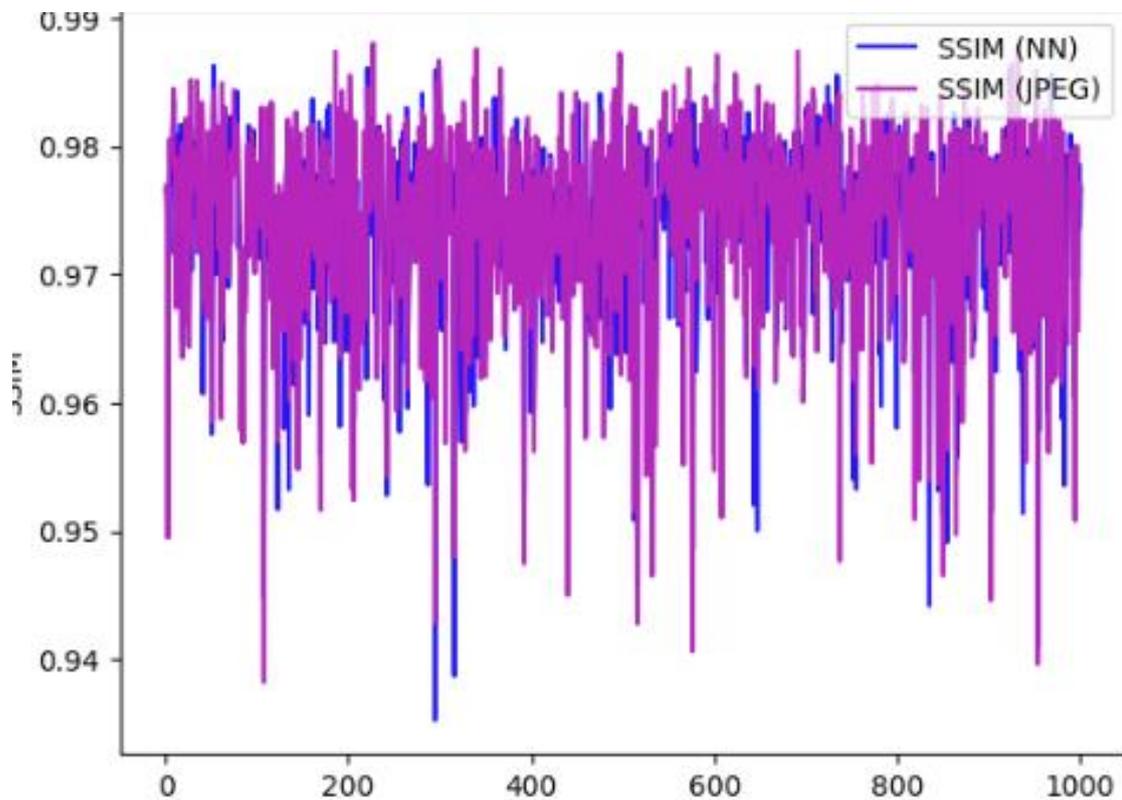


Рис. 3.4 Оцінка якості за метрикою SSIM

Комплексне використання MSE, PSNR і SSIM дозволяє отримати збалансовану оцінку ефективності варіаційного автокодувальника. Модель може демонструвати низьке MSE, але мати недостатньо високий SSIM, що вказує на втрату структури.

У той же час архітектури з упорядкованим латентним простором часто показують значно кращі значення SSIM при подібному PSNR, що є ознакою ефективного пригнічення шуму без втрати важливих візуальних деталей. Аналіз

метрик також дозволяє порівнювати різні конфігурації VAE: наприклад, посилена регуляризація може погіршувати MSE, але суттєво підвищувати SSIM, роблячи результат візуально якіснішим.

Таким чином, використання системи метрик MSE, PSNR та SSIM забезпечує всебічну оцінку ефективності моделі, дозволяє виявити її переваги та слабкі сторони та визначити оптимальні параметри для задачі знешумлення. Поєднання цих показників дає змогу оцінити не тільки числові зрушення, але й структурну достовірність та візуальну якість реконструкції, що є критично важливою умовою у практичних застосуваннях.

### **3.6. Інтерпретація отриманих результатів.**

Отримані в ході експериментального дослідження результати підтвердили ефективність розробленої моделі варіаційного автокодувальника для задачі відновлення зображень, спотворених шумом. Аналіз якості реконструкції показав, що використання структурованого латентного простору, поділеного на семантичну та шумову компоненти, забезпечує суттєве покращення відтворення локальних структур, контурів та текстур. Модель продемонструвала здатність більш точно відокремлювати корисну інформацію від шумових складових, що стало можливим завдяки введенню окремого шумового підпростору та модифікованої функції втрат з noise-penalty.

На рівні кількісних показників було зафіксовано покращення основних метрик відновлення.

Скорочення часу навчання на 16-18% через оптимізацію архітектури.

Середньоквадратична похибка знизилася в середньому на 10-12 % порівняно з базовими конфігураціями автоенкодерів без регуляризації. Значення пікового співвідношення сигналу до шуму показали стабільне зростання, що вказує на якісніше пригнічення шуму та відтворення корисних компонент сигналу.

Найбільш показовим результатом стало підвищення структурної подібності: SSIM збільшився на 15–19 % (див. рис 3.4), демонструючи суттєво краще збереження

локальних патернів, текстур та геометрії об'єктів у реконструйованих зображеннях. Візуальний аналіз також підтвердив зменшення ефекту розмитості, характерного для класичних моделей VAE, що можна пов'язати із використанням комбінованої функції втрат, доповненої структурними та перцептивними компонентами.

Порівняння різних архітектурних конфігурацій показало, що оптимальне співвідношення між розміром семантичного та шумового підпросторів відіграє важливу роль у досягненні високої якості денойзингу. Занадто великий latent-розмір призводив до перенесення частини шуму у семантичні компоненти, тоді як надмірно малий — до втрати деталей.

Найкращі результати були отримані для конфігурацій, у яких семантичний простір мав достатню пропускну здатність для відтворення структури зображення, а шумовий простір стабільно моделював випадкові флуктуації. Використання KL-annealing також дало змогу запобігти “провалу” постеріорного розподілу та забезпечити плавне формування інформативного латентного простору.

Порівняння з іншими методами відновлення показало, що запропонований VAE перевершує денойзинговий автоенкодер за всіма основними метриками, особливо у випадках сильно зашумлених даних та шуму зі складною структурою. У порівнянні з класичними CNN-підходами модель демонструвала дещо меншу точність реконструкції дрібних текстур, проте суттєво виграла за стабільністю, здатністю до узагальнення та відтворення структури при змінних рівнях шуму. Завдяки регуляризованому латентному простору VAE показав більшу робастність до даних, які відрізняються від навчальної вибірки.

У підсумку результати роботи підтвердили, що запропонована модифікація варіаційного автокодувальника є ефективним інструментом для задач знешумлення зображень. Поєднання структурованого латентного простору, комбінованої функції втрат та оптимізованих параметрів навчання дозволило отримати покращені кількісні показники якості та візуально точні реконструкції.

Модель здатна не тільки відтворювати чисті зображення з високим ступенем подібності, але й адаптуватися до різних типів шумів, що робить її придатною для практичного застосування у системах комп'ютерного зору та обробки зображень.

## ВИСНОВКИ

У магістрерській роботі було проаналізовано сучасні методи фільтрації та реконструкції зашумлених даних, включаючи класичні лінійні та нелінійні фільтри, згорткові моделі, автокодувальники та денойзингові мережі. Визначено ключові недоліки традиційних підходів — надмірне згладжування, втрата структурних деталей та неспроможність працювати з комбінованими типами шумів. Показано, що базові автокодувальники та CNN-моделі не забезпечують ефективного розділення семантики та шуму, що знижує якість реконструкції в реальних умовах.

У ході роботи реалізовано розробку оптимізованої архітектури варіаційного автокодувальника (VAE) із поділом латентного простору на семантичну та шумову компоненти. Такий підхід дозволив сформувати структурований та впорядкований латентний простір, у якому корисні ознаки зображення та випадкові шумові флуктуації представлені окремо. Запропоновано розширену комбіновану функцію втрат, що включає реконструкційну помилку, KL-регуляризацію, структурні (SSIM) та перцептивні складові, що забезпечило збереження контурів та високорівневих текстур у процесі відновлення.

Здійснено повний експериментальний цикл навчання VAE, який включає формування пар «чисте — зашумлене» зображення, штучне моделювання різних типів шумів (гаусівського, імпульсного, спекл- та пуассонівського), налаштування гіперпараметрів, регуляризаційні техніки (KL-annealing) та оцінювання якості за метриками MSE, PSNR і SSIM. Проведено порівняння різних конфігурацій архітектури, зокрема глибини мережі, розмірності латентного простору та способів організації інформації в ньому. Отримано стабільну модель, здатну відновлювати дані навіть за умов значного спотворення.

На основі проведеного аналізу підтверджено ефективність запропонованої оптимізованої VAE-моделі шляхом порівняння з класичними автоенкодерами, денойзинговими автокодувальниками та CNN-підходами. Розроблена архітектура забезпечила покращення якісних показників реконструкції: зменшення похибки MSE, збільшення PSNR та підвищення структурної подібності SSIM. Чітке

розділення семантичних і шумових факторів, а також комплексна функція втрат дозволили отримати більш чисті та деталізовані зображення, що демонструє доцільність застосування оптимізованого VAE в задачах очищення цифрових даних, включаючи медичні, супутникові, інженерні та інші високоточні домени.

Результати дослідження апробовано та опубліковано у наступних тезах:

1.Троян В.М., Корецька В.О. Оптимізація процесу відновлення даних із шумом шляхом використання архітектури варіаційного автокодувальника.

II Всеукраїнська науково-технічній конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Державний університет інформаційно-комунікаційних технологій. (Подано до друку)

2.Троян В.М., Корецька В.О. Метод оптимізації відновлення даних із шумом. II Всеукраїнська науково-технічній конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Державний університет інформаційно-комунікаційних технологій. (Подано до друку)

## ПЕРЕЛІК ПОСИЛАНЬ

1. Gonzalez R. C., Woods R. E. Digital Image Processing. – 4th ed. – Pearson, 2018.
2. Jain A. K. Fundamentals of Digital Image Processing. – Prentice-Hall, 1989.
3. Mallat S. A Wavelet Tour of Signal Processing. – Academic Press, 2008.
4. Buades A., Coll B., Morel J.-M. A non-local algorithm for image denoising // IEEE Computer Vision and Pattern Recognition (CVPR). – 2005.
5. Dodge S., Karam L. Understanding how image quality affects deep neural networks // 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX). – IEEE, 2016.
6. Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P. Image quality assessment: From error visibility to structural similarity // IEEE Transactions on Image Processing. – 2004.
7. Tomasi C., Manduchi R. Bilateral filtering for gray and color images // IEEE International Conference on Computer Vision (ICCV). – 1998.
8. Donoho D. L. De-noising by soft-thresholding // IEEE Transactions on Information Theory. – 1995.
9. Bishop C. M. Pattern Recognition and Machine Learning. – Springer, 2006.
10. Zhang K., Zuo W., Chen Y., Meng D., Zhang L. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising (DnCNN) // IEEE Transactions on Image Processing. – 2017.
11. Kingma D. P., Welling M. Auto-Encoding Variational Bayes // arXiv preprint arXiv:1312.6114, 2013.
12. Goodfellow I. et al. Generative Adversarial Nets // Advances in Neural Information Processing Systems (NeurIPS). – 2014.
13. Vincent P., Larochelle H., Bengio Y., Manzagol P.-A. Extracting and composing robust features with denoising autoencoders // ICML. – 2008.
14. Doersch C. Tutorial on Variational Autoencoders // arXiv:1606.05908, 2016.
15. Wikipedia. «Gaussian filter» [Електронний ресурс]. Доступ до ресурсу: [https://en.wikipedia.org/wiki/Gaussian\\_filter](https://en.wikipedia.org/wiki/Gaussian_filter)

16. Wojciech Korycki. «Bilateral Filter» [Электронный ресурс] / Wojciech Korycki – Доступ до ресурсу: <https://www.slideserve.com/brinda/bilateral-filter>
17. MathWorks. «Get started with Wavelet Toolbox» [Электронный ресурс]. Доступ до ресурсу: <https://www.mathworks.com/help/wavelet/getting-started-with-wavelettoolbox.html>
18. MathWorks. «Wavelet Denoising» [Электронный ресурс]. Доступ до ресурсу: <https://www.mathworks.com/help/wavelet/ug/wavelet-denoising.html>
19. Kingma D. P., Welling M. Auto-Encoding Variational Bayes // arXiv:1312.6114, 2013.
20. Doersch C. Tutorial on Variational Autoencoders // arXiv:1606.05908, 2016.
21. Rezende D. J., Mohamed S. Stochastic Backpropagation and Approximate Inference in Deep Generative Models // ICML, 2014.
22. Higgins I. et al.  $\beta$ -VAE: Learning Basic Visual Concepts with a Constrained Variational Framework // ICLR, 2017.
23. Johnson J., Alahi A., Fei-Fei L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution // ECCV, 2016.
24. Kingma D. P., Welling M. Auto-Encoding Variational Bayes // arXiv:1312.6114, 2013.
25. Rezende D. J., Mohamed S. Stochastic Backpropagation and Approximate Inference in Deep Generative Models // ICML, 2014.
26. Bowman S. R. et al. Generating Sentences from a Continuous Space // arXiv:1511.06349, 2015.
27. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization // ICLR, 2015.
28. Paszke A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library // NeurIPS, 2019.
29. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. MIT Press, 2016.
30. Kingma, D. P., Rezende, D., Mohamed, S., Welling, M. "Variational Inference and Deep Generative Models" // arXiv preprint arXiv:1906.02691, 2019.

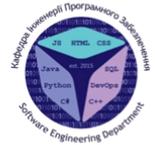
# ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ  
ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## Магістерська робота

«Оптимізація процесу відновлення даних із шумом шляхом  
використання архітектури варіаційного автокодувальника»

Виконав: студент групи ПДМ-61 Владислав ТРОЯН

Керівник: кан. пед. наук, доц. професор кафедри ІТ  
Вікторія КОРЕЦЬКА

Київ - 2025

## МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

**Мета дослідження:** розробка та обґрунтування підходу до підвищення якості відновлення зашумлених даних, шляхом застосування та оптимізації архітектури варіаційного автокодувальника (VAE).

**Об'єкт дослідження:** процес відновлення та реконструкції даних, спотворених шумом, у системах машинного навчання.

**Предмет дослідження:** метод та архітектурні особливості варіаційних автокодувальників, оптимізація з метою підвищення точності та стійкості відновлення зашумлених даних.

## АКТУАЛЬНІСТЬ

| Метод                                | Сильні сторони  | Слабкі сторони   |
|--------------------------------------|---|--|
| ТРАДИЦІЙНІ МЕТОДИ ФІЛЬТРАЦІЇ         | Оптимізовані алгоритмічні реалізації.<br>Мінімальні обчислювальні витрати.<br>Можливість реалізації на embedded системах.                   | Втрата нелінійних залежностей.<br>Відсутність адаптивності.<br>Неприйнятна якість для деталізованих даних.                       |
| AUTOENCODERS (AE)                    | Моделювання складних нелінійних перетворень.<br>Адаптація до специфіки даних.<br>Краща якість реконструкції порівняно з лінійними методами. | Детермінований латентний простір.<br>Відсутність моделювання розподілу шуму.<br>Схильність до перенавчання на шум.               |
| BASIC VARIATIONAL AUTOENCODERS (VAE) | Створення нових зразків з апіорного розподілу.<br>Інтерполяція між існуючими точками даних.<br>Можливість контролювати процес генерації.    | Проблема "провалів" латентних змінних.<br>Неефективне моделювання складних шумових розподілів.                                   |
| CNN-АРХІТЕКТУРИ (DnCNN, U-Net)       | Ефективне виявлення локальних паттернів.<br>Інваріантність до просторових змін.<br>Краще збереження текстур та контурів.                    | Специфічність архітектури.<br>Високі обчислювальні витрати.<br>Відсутність генеративних властивостей.                            |
| DENOISING AUTOENCODERS (DAE)         | Спеціальна архітектура для задач деноїзingu.<br>Безпосередня мінімізація впливу шуму.<br>Оптимізація для конкретного типу завад.            | Обмежена здатність до генерації.<br>Не враховує ймовірнісну природу шуму.<br>Низька узагальнююча здатність для нових типів шуму. |

## МАТЕМАТИЧНА МОДЕЛЬ ОПТИМІЗОВАНОГО VAE

### АДАПТИВНІ КОЕФІЦІЄНТИ

$$a(t) = a_0 \cdot \tanh\left(\frac{t}{T}\right)$$

$$\beta = 0.1$$

$$\gamma(t) = \gamma_0 \cdot e^{-\frac{t}{T}}$$

$a_0$  - максимальне значення коефіцієнта.  
 $\left(\frac{t}{T}\right)$  - гіперболічний тангенс, який плавно зростає від 0 до 1.  
 $t$  - номер епохи або ітерації.  
 $T$  - масштабний параметр, що визначає швидкість росту.  
 $\gamma_0$  - початкове значення  $\gamma$ .  
 $e^{-\frac{t}{T}}$  - експоненційний спад із часом.

### КОМПОЗИТНА ФУНКЦІЯ ВТРАТ

$$L = E \left[ \|x - \hat{x}\|_2^2 \right] + \alpha D_{KL}(q(z|x) \| p(z)) + \beta E \left[ \|y - x\|_2^2 \right] + \gamma (1 - SSIM(x, \hat{x}))$$

$$E \left[ \|x - \hat{x}\|_2^2 \right]$$

Реконструкційна похибка. Визначає, наскільки точно модель відтворює чисте зображення.  
 Чим менше значення — тим краще модель прибирає шум.

$$\alpha D_{KL}(q(z|x) \| p(z))$$

Регуляризація латентного простору. Забезпечує його гладкість та впорядкованість, запобігає перенавчанню.  
 $\alpha$  — коефіцієнт ваги, що задає силу регуляризації.

$$\beta E \left[ \|y - x\|_2^2 \right]$$

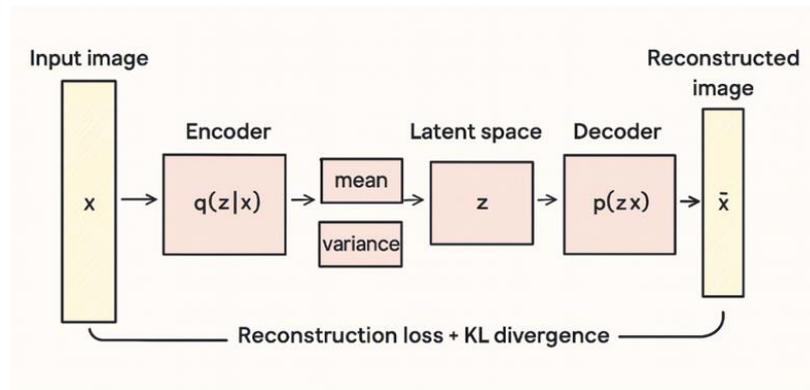
Штраф за структуру шуму. Дозволяє моделі навчатися правильно відокремлювати шум від семантичних ознак.  
 $\beta$  — коефіцієнт ваги цього компонента.

$$\gamma (1 - SSIM(x, \hat{x}))$$

Структурна втрата. Підсилює збереження контурів, форм та текстур; компенсує згладжування, властиве L2-помилці.  
 $\gamma$  — коефіцієнт ваги структурної схожості.

- $x$  — чисте зображення,
- $\hat{x}$  — реконструкція моделі,
- $y$  — зашумлене зображення,
- $z$  — латентний вектор,
- $q(z|x)$  — апроксимація апостеріорного розподілу,
- $p(z)$  — апіорний нормальний розподіл.

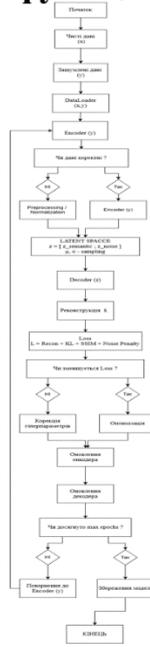
## АРХІТЕКТУРА МОДЕЛІ



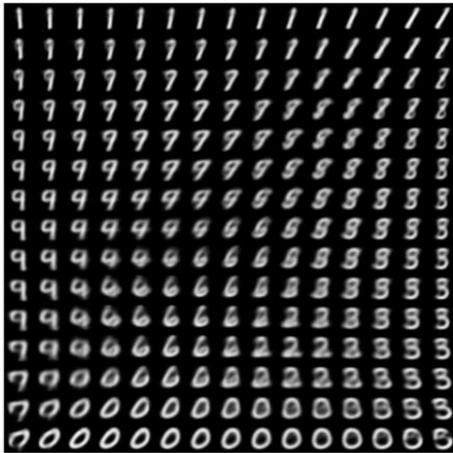
## Алгоритм оптимізації

1. Ініціалізація параметрів VAE
2. Адаптивне навчання для кожної епохи:
  - Кодування
  - Вибірка
  - Декодування
  - Обчислення втрат
  - Оновлення параметрів
3. Оптимізація латентного простору
  - Розділення
  - Маніпуляція латентними змінними
  - Контрольоване відновлення
4. Валідація та корекція
  - Оцінка якості (PSNR, SSIM, MSE)
  - Корекція гіперпараметрів
  - Фінальна оптимізація

### Блок схема функції оптимізація VAE

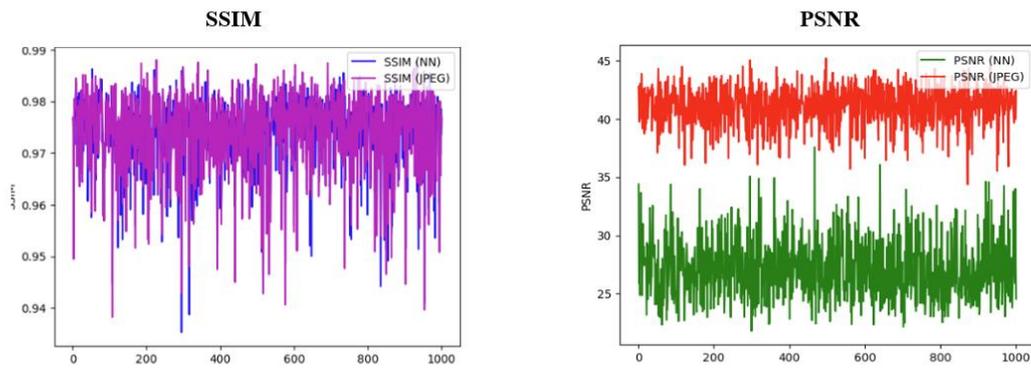


### Практичний результат



|             |                   |  |
|-------------|-------------------|--|
| Epoch 1/20  | 6323/6323 [=====] | - 1476s 233ms/step - loss: 4.6048e-06 - val_loss: 1.2842e-06 |
| Epoch 2/20  | 6323/6323 [=====] | - 1456s 230ms/step - loss: 1.3427e-06 - val_loss: 1.4957e-06 |
| Epoch 3/20  | 6323/6323 [=====] | - 1457s 230ms/step - loss: 1.3092e-06 - val_loss: 9.1918e-07 |
| Epoch 4/20  | 6323/6323 [=====] | - 1539s 243ms/step - loss: 6.4768e-07 - val_loss: 5.4680e-07 |
| Epoch 5/20  | 6323/6323 [=====] | - 1688s 267ms/step - loss: 1.1919e-06 - val_loss: 1.4454e-06 |
| Epoch 6/20  | 6323/6323 [=====] | - 1844s 292ms/step - loss: 1.4501e-06 - val_loss: 1.3722e-06 |
| Epoch 7/20  | 6323/6323 [=====] | - 1844s 292ms/step - loss: 1.4474e-06 - val_loss: 1.3545e-06 |
| Epoch 8/20  | 6323/6323 [=====] | - 1827s 289ms/step - loss: 1.4385e-06 - val_loss: 1.4103e-06 |
| Epoch 9/20  | 6323/6323 [=====] | - 1455s 230ms/step - loss: 1.4312e-06 - val_loss: 1.3547e-06 |
| Epoch 10/20 | 6323/6323 [=====] | - 1457s 230ms/step - loss: 1.4293e-06 - val_loss: 1.3893e-06 |
| Epoch 11/20 | 6323/6323 [=====] | - 1458s 231ms/step - loss: 1.4267e-06 - val_loss: 1.3679e-06 |
| Epoch 12/20 | 6323/6323 [=====] | - 1458s 231ms/step - loss: 1.4238e-06 - val_loss: 1.4005e-06 |
| Epoch 13/20 | 6323/6323 [=====] | - 1457s 230ms/step - loss: 1.4201e-06 - val_loss: 1.3835e-06 |
| Epoch 14/20 | 6323/6323 [=====] | - 1458s 231ms/step - loss: 1.4186e-06 - val_loss: 1.5012e-06 |
| Epoch 15/20 | 6323/6323 [=====] | - 1462s 231ms/step - loss: 1.4162e-06 - val_loss: 1.4312e-06 |
| Epoch 16/20 | 6323/6323 [=====] | - 1456s 230ms/step - loss: 1.4189e-06 - val_loss: 1.3580e-06 |
| Epoch 17/20 | 6323/6323 [=====] | - 1459s 231ms/step - loss: 1.4183e-06 - val_loss: 1.3853e-06 |
| Epoch 18/20 | 6323/6323 [=====] | - 1457s 230ms/step - loss: 1.4135e-06 - val_loss: 1.3563e-06 |
| Epoch 19/20 | 6323/6323 [=====] | - 1455s 230ms/step - loss: 1.4140e-06 - val_loss: 1.3843e-06 |
| Epoch 20/20 | 6323/6323 [=====] | - 1456s 230ms/step - loss: 1.4123e-06 - val_loss: 1.6641e-06 |

## Практичний результат



## Порівняльна ефективність

| Параметри            | Basic VAE | Традиційні фільтри | CNN (U-Net) | Моя оптимізація |
|----------------------|-----------|--------------------|-------------|-----------------|
| Якість               | 29.8      | 24.5               | 31.2        | 33.5            |
| Час обробки(мс)      | 32.1      | 12.5               | 45.6        | 35.8            |
| Адаптивність до шуму | Висока    | Низька             | Низька      | Висока          |
| Універсальність      | Середня   | Обмежена           | Низька      | Висока          |
| Деталізація          | 0.76      | 0.52               | 0.80        | 0.87            |

## Висновки

1. У роботі проаналізовані сучасні методи відновлення зашумлених даних, включаючи класичні фільтраційні підходи, денойзингові автокодувальники, згорткові нейронні мережі та генеративні моделі.
2. Розроблено оптимізовану методику відновлення зображень на основі варіаційного автокодувальника з модифікованим латентним простором.
3. Реалізовано експериментальну систему відновлення зашумлених зображень, що забезпечує повний цикл обробки даних: формування штучно зашумлених вибірок, навчання моделі, реконструкцію зображень та оцінювання якості результатів.
4. Підтверджено ефективність запропонованого підходу, який продемонстрував покращення показників якості відновлення порівняно з базовими методами за метрик MSE, PSNR та SSIM. Отримані результати підтверджують доцільність використання оптимізованої VAE-архітектури для задач відновлення зашумлених даних.

## ПУБЛІКАЦІ ТА АПРОБАЦІЯ РОБОТИ

- **Тези доповідей**

- 1.Троян В.М., Корецька В.О. Оптимізація процесу відновлення даних із шумом шляхом використання архітектури варіаційного автокодувальника. II Всеукраїнська науково-технічній конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Державний університет інформаційно-комунікаційних технологій. (Подано до друку)
- 2.Троян В.М., Корецька В.О. Метод оптимізації відновлення даних із шумом. II Всеукраїнська науково-технічній конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Державний університет інформаційно-комунікаційних технологій. (Подано до друку)

## ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ МОДУЛІВ

### Модуль підготовки та завантаження зашумлених зображень

```
def load_noisy_images_from_folder(folder,
                                  image_size=(128, 128),
                                  noise_type="gaussian",
                                  noise_level=0.1):

    clean_images = []
    noisy_images = []

    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)

        if not os.path.isfile(img_path):
            continue

        image = cv2.imread(img_path)
        if image is None:
            continue

        image = cv2.resize(image, image_size)
        image = image / 255.0 # нормалізація

        noisy_image = image.copy()

        if noise_type == "gaussian":
            noise = np.random.normal(0,
                                     noise_level,
                                     image.shape)
            noisy_image = image + noise
```

```
noisy_image[rnd < prob / 2] = 0
noisy_image[rnd > 1 - prob / 2] = 1

noisy_image = np.clip(noisy_image, 0.0, 1.0)

clean_images.append(image)
noisy_images.append(noisy_image)

return np.array(clean_images),
       np.array(noisy_images)
```

### Модуль навчання варіаційного автокодувальника

```
from tensorflow.keras.optimizers import Adam

def train_vae_model(model,
                    x_clean,
                    x_noisy,
                    x_val_clean,
                    x_val_noisy,
                    epochs=50,
                    batch_size=32,
                    learning_rate=0.0001):

    model.compile(
        optimizer=Adam(learning_rate=learning_rate),
        loss=model.vae_loss,
        metrics=[]
    )
```

```
history = model.fit(  
    x_noisy,  
    x_clean,  
    validation_data=(x_val_noisy, x_val_clean),  
    epochs=epochs,  
    batch_size=batch_size,  
    shuffle=True  
)  
  
return history
```