МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Домбровський Максим Олександрович

Алгоритми синтезу зображень на основі генеративного інтелекту / Image synthesis algorithms based on generative intelligence

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія

Кваліфікаційна робота

Виконав студент групи Кім-21
М.О.Домбровський
_____

Науковий керівник
д.т.н., О.М.Березький
_____

ТЕРНОПІЛЬ- 2024

АНОТАЦІЯ

Домбровський М.О. Алгоритми синтезу зображень на основі генеративного інтелекту. – Рукопис. Кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 123 «Комп'ютерна інженерія», освітньо-професійна програма. Західноукраїнський національний університет, Тернопіль, 2024.

Робота написана обсягом 96 сторінок і містить 18 ілюстрацій, 8 таблиць, 3 додатки та 43 джерела за переліком посилань.

Метою роботи є дослідження дифузійних алгоритмів генерування біомедичних зображень і розроблення узагальненого алгоритму синтезу гістологічних зображень для підвищення якості штучних зображень.

Методи досліджень базуються на використанні математичного аналізу, теорії імовірностей та математичній статистиці, теорії ланцюгів Маркова, теорії нейронних мереж, теорії алгоритмів та методах об'єктно-орієнтованого програмування.

Розроблено узагальнений алгоритм генерування гістологічних зображень. Здійснено програмну реалізацію дифузійних алгоритмів синтезу гістологічних зображень в середовищі Stable Diffusion. Також зроблено оцінку якості синтезованих зображень на основі метрик IS, FID.

Результати роботи можуть бути використані для задач класифікації біомедичних зображень у системах автоматичного діагностування.

Можливими напрямками подальших досліджень є дослідження алгоритмів генерування зображень, які базуються на інших моделях генерування.

КЛЮЧОВІ СЛОВА: СИНТЕЗ, БІОМЕДИЧНІ ЗОБРАЖЕННЯ, ГІСТОЛОГІЧНІ ЗОБРАЖЕННЯ, ДИФУЗІЙНІ МОДЕЛІ, МЕТРИКА FID, МЕТРИКА IS.

ANNOTATION

Dombrovskyi M.O. Algorithms for Image Synthesis Based on Generative Intelligence. – Manuscript. A qualification thesis for obtaining the Master's degree in the specialty 123 "Computer Engineering," educational-professional program. Western Ukrainian National University, Ternopil, 2024.

The thesis is 96 pages in length and includes 18 illustrations, 8 tables, 3 appendices, and 43 references.

The purpose of this work is to investigate diffusion algorithms for biomedical image generation and to develop a generalized algorithm for synthesizing histological images to improve the quality of artificially generated images.

The research methods are based on mathematical analysis, probability theory and mathematical statistics, Markov chain theory, neural networks theory, algorithm theory, and object-oriented programming methods.

A generalized algorithm for generating histological images has been developed. A software implementation of diffusion algorithms for synthesizing histological images was carried out in the Stable Diffusion environment. The quality of the synthesized images was evaluated based on IS and FID metrics.

The results of the work can be applied to biomedical image classification tasks in automatic diagnostic systems.

Possible directions for further research include examining image generation algorithms based on other generative models.

KEYWORDS: SYNTHESIS, BIOMEDICAL IMAGES, HISTOLOGICAL IMAGES, DIFFUSION MODELS, METRIC FID, METRIC IS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

1. IS - Inception Score (Початковий бал).
2. FID - Fréchet Inception Distance (Фреше-відстань у просторі Inception).
3. ГІ – Генеративний інтелект.
4. ШІ- Штучний інтелект.
5. ChatGPT - Generative Pre-trained Transformer.
6. МРТ - Магнітно-резонансна томографія.
7. УЗД - Ультразвукове дослідження.
8. ГЗМ - Генеративні змагальні мережі.
9. ЗНМ - Згорткові нейронні мережі.
10. ВА - Варіаційні автокодувальники.
11. UI - User Interface (Інтерфейс користувача).
12. API - Application Programming Interface (інтерфейс прикладного програмування).
13. DM - Diffusion Models (Дифузійні моделі).
14. LDM - Latent Diffusion Models (Модель латентної дифузії).
15. LMU - Ludwig-Maximilians-Universität München (Людвіг-Максиміліанівський університет Мюнхена).
16. CLIP - Contrastive Language-Image Pre-training (Підготовка до контрастних мовно-образних тренінгів).
17. GPU - Graphics Processing Unit (Графічний процесор).
18. VRAM - Video Random Access Memory (Відеопам'ять з довільним доступом).
19. PNG - Portable Network Graphics (Портативна мережева графіка).
20. DPM++ - Denoising Probabilistic Model ++ (Покращена ймовірнісна модель денойзінгу).
21. LMS - Laplacian Multi-Step Sampling (Багатокроковий семплінг із використанням лапласіанів).

22. DDIM - Denoising Diffusion Implicit Models (Імпліцитна модель дифузії денойзінгу).

23. CFG Scale - Classifier-Free Guidance Scale (Масштаб керування без класифікатора).

24. ReLU - Rectified Linear Unit (Випрямлений лінійний блок).

25. ID – Identifier (Ідентифікатор).

26. GB – Gigabyte (Гігабайт).

ВСТУП

Актуальність дослідження. Класифікація біомедичних зображень є актуальною проблемою систем автоматичного діагностування. Сучасні засоби класифікації будуються на основі глибоких нейронних мереж. Ці потужні засоби вимагають великих навчальних вибірок. Наявні навчальні вибірки є обмеженими через об’єктивні причини. Тому єдиним виходом із даної ситуації є генерування штучних зображень на основі обмежених реальних вибірок. Для цього використовують методи та програмні засоби генеративного інтелекту, який на даний час є вершиною досягнень штучного інтелекту. Використання дифузійних моделей і програмних засобів, які побудовані на їх основі, для генерування біомедичних зображень є перспективним і актуальним напрямком генеративного інтелекту. Тому дане дослідження є актуальним для задач генерування біомедичних зображень.

Мета і завдання дослідження. Метою кваліфікаційної роботи є дослідження дифузійних алгоритмів генерування біомедичних зображень і розроблення узагальненого алгоритму синтезу гістологічних зображень для підвищення якості штучних зображень.

У відповідності із поставленою метою кваліфікаційна робота включає розв’язки таких задач:

– проаналізувати напрямки досліджень у штучному інтелекті;

– провести аналіз гістологічних зображень;

– провести аналіз підходів до синтезу зображень на основі генеративного інтелекту;

– проаналізувати програмні засоби синтезу зображень;

– дослідити дифузійні алгоритми генерування зображень;

– розробити узагальнений алгоритм генерування гістологічних зображень;

– провести обчислювальні експерименти для синтезу гістологічних зображень і оцінити якість синтезованих зображень на основі метрик IS, FID.

Об’єкт дослідження – процес синтезу біомедичних зображень.

Предмет дослідження – дифузійні алгоритми синтезу зображень.

Методи досліджень. У кваліфікаційній роботі використано математичний аналіз, теорію імовірностей та математичну статистику, теорію ланцюгів Маркова, теорію нейронних мереж, теорію алгоритмів та методи об’єктно-орієнтованого програмування.

Наукова новизна одержаних результатів полягає в дослідженні дифузійних алгоритмів генерування зображень, розроблені узагальненого алгоритму генерування гістологічних зображень.

Практичне значення одержаних результатів полягає в проведені обчислювальних експериментів в середовищі Stable Diffusion для синтезу гістологічних зображень заданої якості.

Публікації результатів досліджень. За результатами досліджень опубліковані три тези доповідей Всеукраїнської науково-практичної конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі» (5 грудня 2023 р. та 21 травня 2024 р., м. Тернопіль, Західноукраїнський національний університет), а також матеріали міжнародної конференції [1, 2, 3]:

1. Домбровський М. О. Синтез зображень на основі генеративного інтелекту. *Інтелектуальні комп’ютерні системи та мережі* : тези доп. VIII Наук.-практ. конф. молодих вчених і студентів (5 груд. 2023 р.). Тернопіль : ЗУНУ, 2023. С. 28.

2. Домбровський М. О. Синтез біомедичних зображень на основі дифузійно – імовірнісних моделей. *Інтелектуальні комп’ютерні системи та мережі* : тези доп. IX Наук.-практ. конф. студентів, аспірантів, молодих вчених (21 трав. 2024 р.). Тернопіль : ЗУНУ, 2024. С. 34.

3. Berezsky O., Liashchynskyi P., Melnyk G., Dombrovskyi M. and Berezkyi M. Synthesis of biomedical images based on generative intelligence tools. 7th International Conference on Informatics & Data-Driven Medicine. University of

Birmingham, Birmingham, United Kingdom. November 14 - 16, 2024. (Scopus, Web of Science).

Кваліфікаційна робота складається із трьох розділів, висновків, списку використаної літератури та додатків [4].

В першому розділі показано актуальність досліджень в генеративному інтелекті, проаналізовано підходи до генерування зображень, проаналізовані програмні засоби генеративного інтелекту для генерування зображень.

В другому розділі досліджено дифузійні алгоритми синтезу зображень, розроблено узагальнений алгоритм синтезу гістологічних зображень.

В третьому розділі описано програмне середовище Stable Diffusion, описано обчислювальні експерименти для синтезу гістологічних зображень і проведено їх оцінку на основі метрик IS, FID.

У додатку приведено світлокопії виданих публікацій та аугментована вибірка гістологічних зображень.

1 АНАЛІЗ МЕТОДІВ АЛГОРИТМІВ І ПРОГРАМНИХ ЗАСОБІВ СИНТЕЗУ ЗОБРАЖЕНЬ

1.1 Поняття та особливості генеративного інтелекту

Генеративний інтелект (ГІ) у сфері зображень відіграє ключову роль у сучасному світі штучного інтелекту (ШІ), пропонуючи широкий спектр можливостей для створення, модифікації та аналізу різноманітного контенту.

Аналіз звіту Gartner Hype Cycle™ за 2023 рік підкреслює значний вплив інновацій у галузі штучного інтелекту. Генеративний штучний інтелект, зокрема системи типу ChatGPT, значно підвищив продуктивність розробників і спеціалістів, змусивши багато організацій переосмислити цінність людських ресурсів та свої бізнес-моделі. На рисунку 1.1 приведено візуалізацію технологічних трендів та їх прогнозований вплив на бізнес і індустрію. Проаналізуємо криву розвитку технологій ШІ.

1. ШІ – це гіпотетичний рівень машинного інтелекту, що дозволяє машині виконувати інтелектуальну діяльність, яку може здійснювати людина.
2. Інженерія ШІ – це фундаментальна дисципліна, що забезпечує розробку, доставку та операційне використання ШІ на підприємствах.
3. Автономні системи – самокеровані системи, що виконують обмежені завдання і характеризуються автономністю, навчанням та володінням ініціативою.
4. Хмарні ШІ-послуги – інструменти для побудови ШІ-моделей, API для готових сервісів, що дозволяють розгортати та використовувати машинне навчання на хмарній інфраструктурі.
5. Композитний ШІ – це поєднання різних технік ШІ для покращення ефективності навчання та збільшення рівня представлення знань.
6. Комп'ютерний зір – це технології, що аналізують реальні зображення та відео для вилучення значущої інформації.

7. Периферійний ШІ – використання ШІ у не ІТ-продуктах, пристроях IoT, шлюзах, що можуть використовуватися в споживчих, комерційних та промислових цілях.

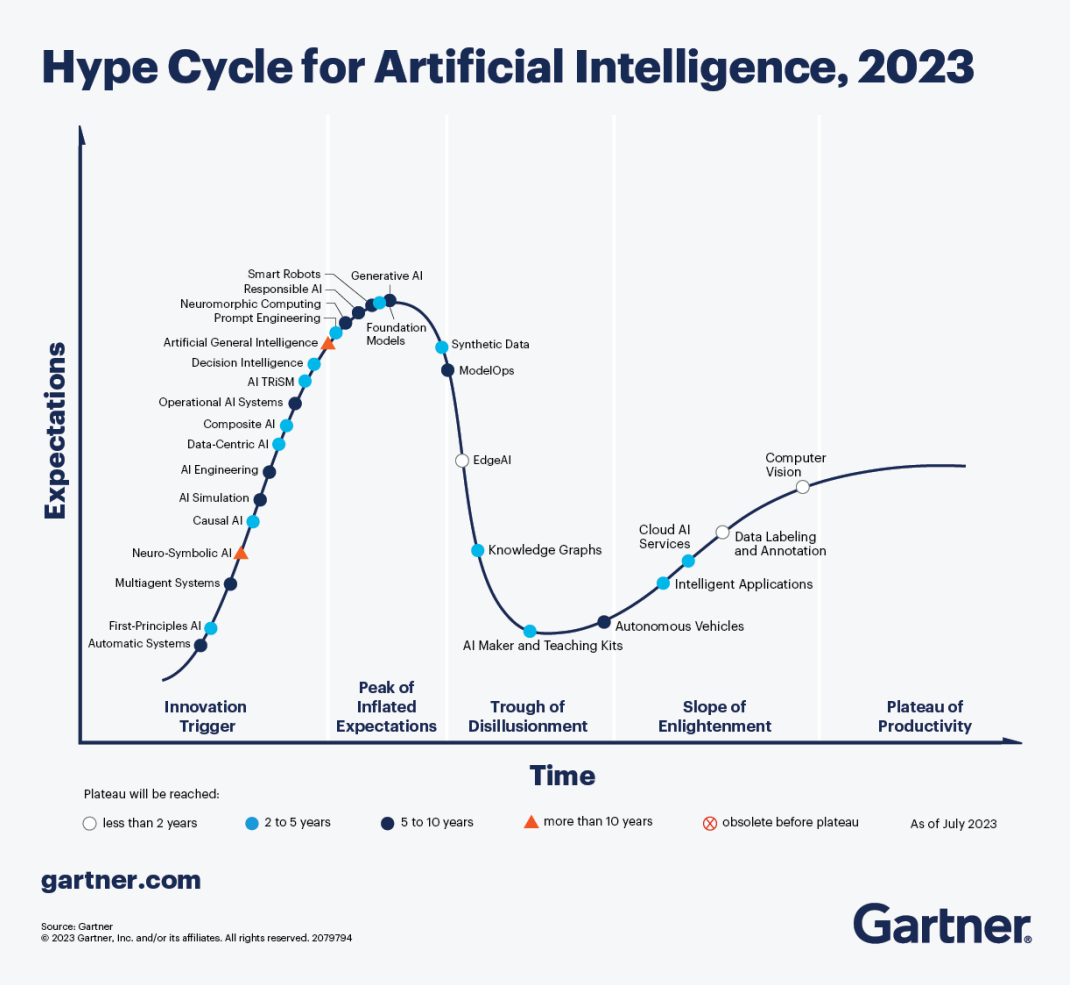


Рисунок 1.1 – Візуалізація ключових технологічних трендів зі звіту Gartner Hype Cycle за 2023 рік

Як видно із цієї кривої ГІ на даний час є на піку розвитку ШІ. Розглянемо застосування ГІ для задач синтезу зображень.

ГІ використовує алгоритми машинного навчання та глибокого навчання для генерації нових зображень або внесення змін у існуючі зображення, заснованих на навчених моделях даних. Основні характеристики цієї технології такі:

1. Реалістичність: ГІ може створювати зображення, які візуально невідрізнені від справжніх фотографій, забезпечуючи високу ступінь деталізації та точності.

2. Творчість: Генеративні моделі відкривають нові можливості для художників та дизайнерів, дозволяючи експериментувати з унікальними стилями та естетикою.

3. Адаптивність: Системи ГІ можуть бути навчені для генерації зображень в різних стилях та контекстах, реагуючи на змінні вимоги та уподобання користувачів.

4. Інтерактивність: Користувачі можуть взаємодіяти з ГІ, вносячи зміни у параметри генерації або надаючи власні вхідні дані для отримання індивідуалізованих результатів.

Розглянемо приклади застосування ГІ. В основному ГІ застосовують для таких задач:

1. Генерація арт-зображень. ГІ може створювати нові художні твори, імітуючи різні стилі та техніки, або генерувати абстрактні візуальні композиції.

2. Синтез облич. Технологія використовується для створення реалістичних зображень облич, які можуть бути застосовані в галузі розваг, безпеки та цифрової анімації.

3. Покращення зображень. ГІ використовується для автоматичного покращення якості зображень, включаючи видалення шуму, підвищення роздільної здатності та корекцію кольору.

4. Заповнення відсутніх частин. ГІ може відновлювати втрачені або пошкоджені частини зображень, що є корисним у реставрації старих фотографій або видаленні небажаних об’єктів.

5. Перетворення стилю. За допомогою ГІ можна змінити стиль зображення, наприклад, перетворити фотографію в картину, виконану в стилі відомого художника.

6. Анімація облич. Генеративні моделі можуть анімувати обличчя на зображеннях або відео, створюючи ефект розмови або емоційних виразів.

7. Біомедицина. Генеративні моделі використовуються для синтезу біомедичних зображень [5-7].

Розглянемо детальніше застосування ГІ в біомедицині.

У задачі синтезу біомедичних зображень ГІ генерує реалістичні медичні зображення для навчання та досліджень, дозволяючи створювати додаткові дані без необхідності проведення додаткових медичних процедур. Для задачі діагностики генеративні моделі в автоматичному режимі виявляють та класифікують патології на медичних зображеннях, таких як рентгенівські знімки, МРТ або УЗД. Для задачі відновлення зображень ГІ використовується для покращення якості медичних зображень, наприклад, шляхом видалення шуму або підвищення роздільної здатності, що може поліпшити точність діагностичних висновків. Для задачі моделювання анатомічних структур генеративні моделі використовуються для створення деталізованих 3D-моделей анатомічних структур, які можуть використовуватися в хірургічному плануванні або медичному навчанні.

1.2 Аналіз біомедичних зображень на прикладі гістологічних зображень

Аналіз біомедичних зображень є важливою частиною медичних досліджень і діагностики, дозволяючи лікарям і науковцям вивчати структуру та функції клітин, тканин і органів [8-9]. Гістологічні зображення, які показують детальну структуру тканин на клітинному рівні, є одним із найцінніших видів зображень у біомедицині, особливо в онкології, патології та інших спеціалізованих галузях медицини. Гістологічні зображення дозволяють оцінити структуру тканин, виявляти патологічні зміни та визначати мікрооточення клітин, що є важливим для точної діагностики та вибору ефективних методів лікування. Вони дають змогу детально вивчати складні структурні зміни, такі як

атипові клітини, метастази та мікроскопічні ураження, що особливо цінне в онкологічній практиці. Реальні гістологічні зображення показано на рисунку 1.2.

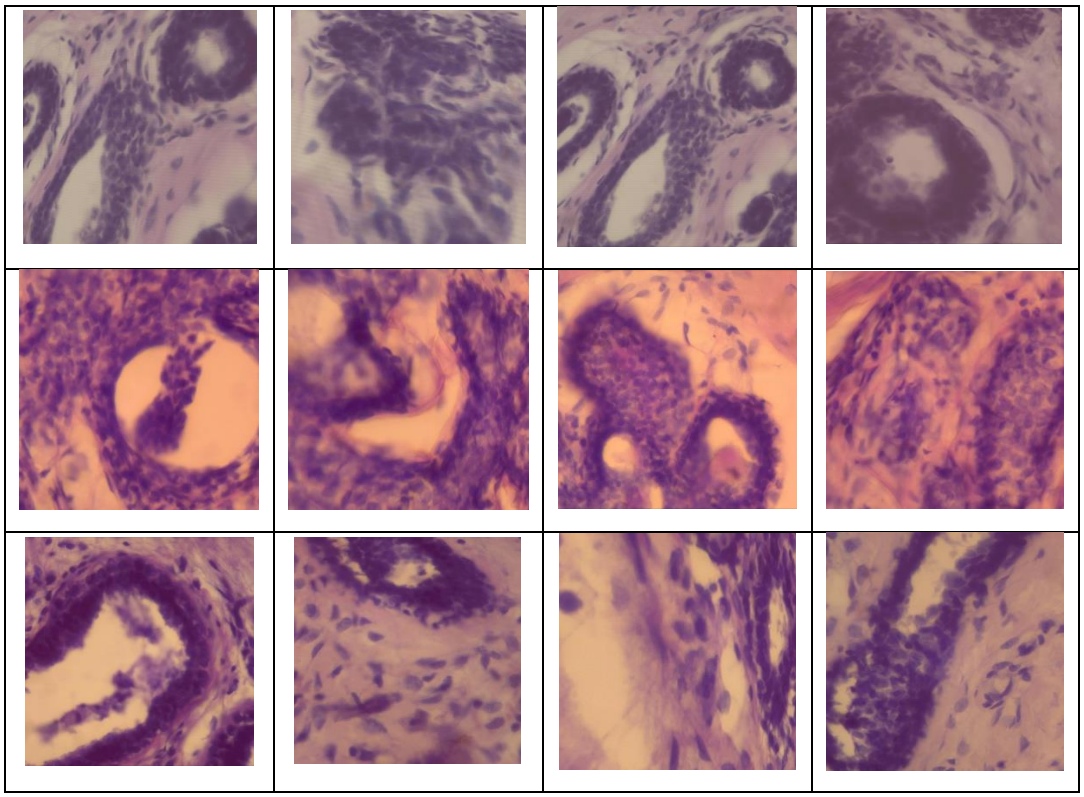


Рисунок 1.2 – Реальні гістологічні зображення

Мастопатія є одним із найпоширеніших захворювань молочних залоз у жінок, що спостерігається у значного відсотка репродуктивного віку. Це захворювання може мати як доброякісний характер, так і підвищувати ризик розвитку злоякісних новоутворень, що робить його важливим об’єктом для ранньої діагностики та моніторингу. Для автоматизації діагностики патологічних станів молочної залози важливо розуміти, що медичні поняття, котрі описують стани відображаються в структурні та текстурні особливості гістологічних зображень.

Набір даних для тренування моделей складається з цифрових гістологічних зображень в bmp, jpg форматах, отриманих з використанням фарбування гематоксиліном та еозином. Зображення зберігаються у високій роздільній здатності 3 664 х 2 748 пікселів, що дозволяє розрізняти тонкі деталі тканинної

структури. Зображення датасету поділені на три класи: Фіброзно-кістозна мастопатія, Непроліферативна мастопатія, Проліферативна мастопатія.

Гістологічні зрізи тканини молочної залози, що використовуються для аналізу мастопатії, демонструють складну будову, що складається з паренхіми та строми. Паренхіма представлена альвеолами та протоками, які формують функціональну частину тканини. Аальвеоли виглядають як дрібні округлі або овальні структури, оточені шаром епітеліальних клітин, з чіткими межами. Строма утворена сполучною тканиною, що забезпечує механічну підтримку та кровопостачання паренхіми. При аналізі гістологічних зрізів для діагностики мастопатії, особливо важливо звертати увагу на зміни структури паренхіми, щільності та складу строми, а також на можливі патологічні утворення, такі як кісти чи проліферативні ділянки. Термін "проліферація" часто використовується для опису надмірного поділу клітин, що може призводити до потовщення стінок протоків молочної залози, змін у структурі альвеол або розростання сполучної тканини. На зображення проліферативні зони характеризуються підвищеною клітинною щільністю.

Фіброзно-кістозна мастопатія характеризується надмірним розростанням фіброзної тканини, що призводить до ущільнення строми, а також появою кістозних утворень у протоках молочної залози. Кістозні утворення- це порожнини з чіткими межами, заповнені рідиною, які можуть утворюватися у протоках молочної залози. Фіброзні ділянки характеризуються щільною текстурою і однорідністю, що також може бути використано для навчання моделей.

Непроліферативна мастопатія має відмінності, які полягають у відсутності активного клітинного поділу. Структура протоків і альвеол зберігається без значних змін, а фіброзна тканина демонструє стабільність. Такі гістологічні зображення часто характеризуються слабкою контрастністю між здоровими і патологічно зміненими ділянками. Це створює додаткові труднощі для автоматичного сегментування, оскільки моделі потребують тонкого аналізу текстури тканин для правильної ідентифікації навіть незначних змін.

Проліферативна мастопатія супроводжується активним поділом клітин, що веде до потовщення стінок протоків та альвеол, а також до появи атипії клітин. У стромі спостерігається збільшення судинної сітки та гетерогенність тканинної структури. На гістологічних зображеннях ці зміни проявляються як різноманітні текстурні та морфологічні зміни, що ускладнює сегментацію [10-11].

Кожне зображення супроводжується вручну створеними анотаціями, які визначають межі патологічних зон, включаючи кістозні утворення, фіброзну тканину, проліферативні ділянки та здорові зони. Анотації виконуються експертами в галузі гістології, що гарантує точність маркування. Це дозволяє навчити модель точному розпізнаванню та сегментації різних структур.

Попередня обробка зображень включає нормалізацію кольорової гами для зменшення впливу варіацій у фарбуванні. Також застосовується аугментація, яка включає поворот, зміну яскравості, контрасту та масштабування для підвищення стійкості моделі до варіацій.

1.3 Аналіз підходів до генерування зображень

Проаналізуємо основні підходи, які використовуються для генерування зображень. Стохастичні диференційні рівняння є важливим математичним інструментом у моделюванні випадкових процесів, особливо в системах, де шум і флуктуації мають суттєве значення [12]. Для генеративного штучного інтелекту та моделей, таких як Stable Diffusion, ці рівняння використовуються для опису процесів генерації даних через моделювання додавання та видалення шуму. У генеративних моделях штучного інтелекту стохастичні диференційні рівняння застосовуються для перетворення розподілу гаусового шуму у складні розподіли даних, такі як зображення чи текст. Під час навчання модель поступово додає шум до даних, моделюючи цей процес за допомогою стохастичних рівнянь.

Модель навчається відновлювати початкові дані з шуму, освоюючи зворотний процес видалення шуму.

Stable Diffusion є прикладом генеративної моделі, яка використовує дифузійні процеси для створення високоякісних зображень. Під час навчання до зображень додається шум, і модель навчається передбачати, як його видалити. При генерації нових зображень процес починається з випадкового шуму, і модель поступово зменшує рівень шуму, генеруючи реалістичне зображення. Це досягається завдяки використанню стохастичних диференційних рівнянь, які забезпечують математичну строгість і стабільність процесу генерації.

Використання стохастичних диференційних рівнянь у генеративних моделях має кілька переваг. По-перше, вони надають математичну основу для розуміння та аналізу генеративних процесів. По-друге, дозволяють моделювати складні розподіли даних за допомогою стохастичних процесів. По-третє, забезпечують високу якість генерації, оскільки моделі на основі цих рівнянь можуть створювати реалістичний контент з високим ступенем деталізації.

Дифузійні моделі є сучасним підходом у галузі генеративного штучного інтелекту, який використовує принципи дифузійних процесів для створення нових даних, таких як зображення, текст або звук [13-15]. Ці моделі стали популярними завдяки своїй здатності генерувати високоякісний та реалістичний контент, що перевершує попередні генеративні моделі, такі як варіаційні автоенкодери або генеративні змагальні мережі.

Дифузійні моделі базуються на концепції поступового додавання шуму до даних та їхнього відновлення [16-17]. Процес складається з двох етапів:

1. Прямий процес (forward process): Початкові дані, наприклад зображення, поступово зашумлюються протягом певної кількості кроків. Це можна уявити як процес дифузії, де інформація розсіюється та стає все менш впорядкованою.

2. Зворотний процес (reverse process): Модель навчається відновлювати початкові дані з шуму, виконуючи зворотний дифузійний процес. Метою є поступове видалення шуму та реконструкція оригінальних даних.

Дифузійні моделі працюють у двох режимах [18-19]:

1. Навчання. Під час навчання модель спостерігає за тим, як незашумлені дані переходять у шумові, і навчається передбачати розподіл ймовірностей для зворотного процесу на кожному кроці.

2. Генерація. Для генерації нових даних модель починає з випадкового шуму і поступово застосовує зворотний процес, створюючи нові зразки, які відповідають розподілу навчальних даних.

Крім цих підходів використовуються інші засоби для опрацювання зображень [12]. Розглянемо деякі із них:

1. Генеративні змагальні мережі (ГЗМ): Використовуються для створення реалістичних зображень за допомогою змагання між генератором, що створює зображення, та дискримінатором, який їх оцінює [21].

2. Автокодувальники: Нейронні мережі, що складаються з енкодера та декодера, які перетворюють вхідні зображення у компактне внутрішнє представлення і відтворюють їх знову, використовуються для зменшення розмірів, видалення шумів або зміни стилю [22].

3. Згорткові нейронні мережі (ЗНМ): Ефективно виявляють шаблони і особливості на зображеннях, використовуються для класифікації, розпізнавання об'єктів і багато іншого.

4. Трансформери: Використовують механізм уваги для ефективної обробки великих обсягів інформації і показали вражаючі результати в генерації тексту та зображень.

5. Варіаційні автокодувальники (ВА): Використовують варіаційне баєсове навчання для генерації нових зображень, моделюючи розподіл вхідних даних.

6. Моделі змішаного навчання: Комбінують елементи різних підходів, як ГЗМ та автокодувальники, для поліпшення якості генерованих зображень.

7. Глибоке посилене навчання: Застосовується для оптимізації процесу створення зображень, де моделі навчаються вибирати найкращі дії на основі отриманих винагород [23-24].

8. Нейронні стилізації: Техніка, що дозволяє переносити стиль одного зображення на інше, оптимізуючи функцію втрат, що враховує відмінності у стилі та збереження вмісту [25-28].

9. Адаптація до конкретних доменів: Вимагає налаштування моделей з урахуванням особливостей даних у певній області, включаючи використання спеціалізованих архітектур мереж або методів попереднього навчання.

10. Оптимізація гіперпараметрів: Включає налаштування швидкості навчання, розміру пакету, кількості шарів та інших параметрів для досягнення найкращих результатів у генерації зображень.

11. Семантичне моделювання: Зосереджене на розумінні та генерації зображень з урахуванням їх семантичного змісту, використовуючи глибоке навчання для аналізу семантичних відносин між об'єктами.

1.4 Аналіз програмних засобів генеративного інтелекту

Проаналізуємо чотири різних інструменти для генерації зображень на основі ГІ: Stable Diffusion UI, Comfy UI, DALL-E, та MidJourney. Кожен з цих інструментів має свої унікальні характеристики, які визначають їх придатність для різних сфер застосування та задач. Виділимо основні критерії для порівняння програмних засобів : доступність, основне використання, сфера застосування, кастомізація, обчислювальні ресурси, особливості та можливість навчання.

Проведемо порівняння програмних засобів на основі цих критеріїв.

Доступність. Stable Diffusion UI та Comfy UI пропонують моделі з відкритим кодом, що дозволяє користувачам модифікувати та розширювати їх відповідно до своїх потреб. DALL-E доступний через API з закритим кодом, що обмежує можливості внесення змін у саму модель, але дозволяє використовувати потужні алгоритми OpenAI. MidJourney є інструментом з обмеженим доступом, що в основному орієнтований на арт-проекти та креативний дизайн.

Основне використання та сфера застосування. Stable Diffusion UI та Comfy UI підтримують широкий спектр застосувань, включаючи генерацію зображень та інтеграцію різних AI моделей, що робить їх універсальними інструментами для різноманітних завдань. DALL-E фокусується на мистецтві, пропонуючи інноваційні алгоритми для створення високоякісних зображень з текстових описів. MidJourney спеціалізується на арт-проектах та креативному дизайні, вирізняючись своїм унікальним стилем та якістю зображень.

Кастомізація та обчислювальні ресурси. Stable Diffusion UI надає високу кастомізацію з можливістю долаштовування та навчання моделі на основі власних зображень. Comfy UI, залежно від платформи, може мати обмеження щодо кастомізації. DALL-E та MidJourney мають обмежені можливості кастомізації, зосереджуючись на стандартних параметрах генерації зображень.

Особливості та можливість навчання. Stable Diffusion UI та Comfy UI підтримують інтеграцію і модифікацію, що дозволяє використовувати їх для широкого спектру завдань. Stable Diffusion UI також дозволяє навчання моделі на основі своїх зображень. DALL-E та MidJourney не дозволяють навчання на основі своїх зображень, що може бути обмеженням для деяких користувачів.

Порівняння програмних засобів приведено у таблиці 1.1.

Деталізуємо ці програмні засоби.

Stable Diffusion - це модель глибокого навчання, що перетворює текст на зображення, розроблена у 2022 році на основі дифузійних технік. використовується в основному для генерації деталізованих зображень на основі текстових описів. Крім того, модель може застосовуватися для таких завдань, як інпейнтинг, аутпейнтинг та генерація перекладів зображення в зображення на основі текстового запиту.

Розробка Stable Diffusion велася дослідниками з групи CompVis у Людвіг-Максиміліанському університеті Мюнхена та компанії Runway з обчислювальним внеском від Stability AI та навчальними даними від некомерційних організацій. Код та ваги моделі були відкриті для загального

доступу, що дозволяє її використовувати на більшості споживчого обладнання, оснащеного помірним графічним процесором з мінімум 4 ГБ відеопам'яті.

Stable Diffusion використовує архітектуру дифузійної моделі (DM), зокрема, модель латентної дифузії (LDM), розроблену групою CompVis у LMU Мюнхен.

Таблиця 1.1 – Порівняння програмних засобів

Критерії	Stable Diffusion UI	Comfy UI	DALL-E	MidJourney
1	2	3	4	5
Доступність	Відкритий код	Відкритий код	API доступ, код не відкритий	Пропрієтарний, обмежений доступ
Основне використання	Генерація зображень. Графічний інтерфейс для AI	Генерація зображень. Графічний інтерфейс для AI	Генерація зображень	Генерація зображень
Сфера застосування	Широкий спектр. Підтримка різних AI моделей	Широкий спектр. Підтримка різних AI моделей	Широкий спектр, акцент на мистецтві	Арт-проекти, креативний дизайн
Кастомізація	Є можливість тонкої настройки моделі	Залежить від платформи	Обмежена	Обмежена
Обчислювальні ресурси	Високі вимоги	Високі вимоги	Високі вимоги	Високі вимоги

Продовження таблиці 1.1

1	2	3	4	5
Особливості	Висока якість зображень. Підтримка спільноти, модифікація. Інтерфейс користувача, інтеграція	Підтримка спільноти, модифікація. Інтерфейс користувача, інтеграція	Інноваційні алгоритми	Висока якість зображень, унікальний стиль
Можливість навчання на основі своїх зображень	Так (можливість донастроювання та навчання моделі)	Ні	Ні, доступ лише через API з певними обмеженнями	Обмежена, переважно стандартні параметри стилів

Дифузійні моделі, запропоновані у 2015 році, навчаються з метою послідовного видалення гаусового шуму з навчальних зображень [29-31]. Stable Diffusion складається з трьох частин: варіаційного автокодувальника (VAE), U-Net та необов'язкового текстового кодера. VAE стискає зображення з піксельного простору до меншого вимірного латентного простору, захоплюючи більш фундаментальне семантичне значення зображення. На етапі прямої дифузії до стислого латентного представлення ітеративно застосовується гаусовий шум. Блок U-Net, який складається з основи ResNet, видаляє шум від виходу прямої дифузії назад, щоб отримати латентне представлення. Нарешті, декодер VAE генерує кінцеве зображення, перетворюючи представлення назад у піксельний простір. Крок знешумлення може бути гнучко визначений текстом, зображенням або іншою модальністю. Кодовані дані умов вводяться в U-Net за допомогою механізму перехресної уваги. Для умовної генерації на основі тексту використовується фіксований попередньо навчений текстовий кодер CLIP ViT-L/14, який перетворює текстові підказки в простір вбудовування. Однією з

переваг LDM є підвищена обчислювальна ефективність для навчання та генерації. З 860 мільйонами параметрів в U-Net і 123 мільйонами в текстовому кодері. Stable Diffusion вважається відносно легкою моделлю за стандартами 2022 року і, на відміну від інших дифузійних моделей, може працювати на користувацьких GPU.

Розширена версія Stable Diffusion SD XL використовує ту саму архітектуру, але збільшену: більший U-Net, більший контекст перехресної уваги, два текстових кодери замість одного, і навчання на кількох співвідношеннях сторін (не лише квадратному, як у попередніх версіях). Версія 3.0 Stable Diffusion повністю змінює основу. Замість U-Net використовується Rectified Flow Transformer, який реалізує метод виправленого потоку з архітектурою Transformer.

Stable Diffusion була навчена на парах зображень та підписів, взятих з LAION-5B, публічно доступного набору даних, похідного від даних Common Crawl, веб-скребінгу, де 5 мільярдів пар зображень-текстів були класифіковані на основі мови та відфільтровані в окремі набори даних за роздільною здатністю.

Модель спочатку навчалася на підмножинах laion2B-en та laion-high-resolution, а останні раунди навчання проводилися на LAION-Aesthetics v2 5+, підмножині 600 мільйонів підписаних зображень. Навчання моделі проводилося за допомогою 256 графічних процесорів Nvidia A100 на Amazon Web Services протягом 150 000 годин-GPU, що коштувало $600 000.

Stable Diffusion має проблеми з неточностями в певних сценаріях. Початкові версії моделі навчалися на наборі даних, який складається з зображень роздільною здатністю 512×512, що означає, що якість генерованих зображень помітно погіршується, коли специфікації користувача відрізняються від "очікуваної" роздільної здатності 512×512. Оновлення версії 2.0 моделі Stable Diffusion пізніше ввело можливість генерувати зображення з рідною роздільною здатністю 768×768. Інша проблема пов'язана з генерацією людських кінцівок через низьку якість даних про кінцівки в базі даних LAION. Модель недостатньо навчена розуміти людські кінцівки та обличчя через відсутність даних.

Для індивідуальних розробників доступність також може бути проблемою. Для налаштування моделі під нові випадки використання, які не включені в набір даних, такі як генерація аніме-персонажів ("waifu diffusion"), потрібні нові дані та подальше навчання. Адаптації Stable Diffusion, створені шляхом додаткового повторного навчання, використовуються для різноманітних випадків використання, від медичної візуалізації до алгоритмічно генерованої музики. Однак цей процес налаштування чутливий до якості нових даних; зображення низької роздільної здатності або різні роздільності від оригінальних даних не тільки можуть не навчитися новому завданню, але й погіршити загальну продуктивність моделі. Навіть коли модель додатково навчається на зображеннях високої якості, індивідуальним користувачам важко запускати моделі на користувацьких апаратних засобах. Наприклад, процес навчання для "waifu-diffusion" вимагає мінімум 30 ГБ VRAM, що перевищує звичайні ресурси, надані такими споживчими GPU, як серія Nvidia GeForce 30, яка має лише близько 12 ГБ.

Творці Stable Diffusion визнають потенціал алгоритмічної упередженості, оскілки модель була переважно навчена на зображеннях з англійськими описами. В результаті генеровані зображення посилюють соціальні упередження. Модель надає більш точні результати для підказок, написаних англійською мовою, порівняно з тими, що написані іншими мовами, причому західна або біла культура часто є стандартним представленням.

Для усунення обмежень початкового навчання моделі, кінцеві користувачі можуть використовувати додаткове навчання для тонкого налаштування результатів генерації, щоб вони відповідали більш конкретним випадкам використання. Існують три методи, за допомогою яких користувачі можуть застосовувати налаштування до контрольної точки моделі Stable Diffusion:
Тренування "вбудовування" (embedding): Вбудовування можуть бути навчені на основі колекції зображень, наданих користувачем, і дозволяють моделі генерувати візуально схожі зображення, коли назва вбудовування використовується в запиті на генерацію.

Гіпермережа (hypernetwork): Гіпермережа - це невелика попередньо навчена нейронна мережа, яка застосовується до різних точок у більшій нейронній мережі. Гіпермережі спрямовують результати в певному напрямку, дозволяючи моделям на основі Stable Diffusion імітувати художній стиль певних художників. DreamBooth: DreamBooth - це модель глибокого навчання, яка може тонко налаштовувати модель для генерації точних, персоналізованих результатів, які зображують конкретний предмет, після навчання за допомогою набору зображень, на яких зображений предмет.

Модель Stable Diffusion підтримує здатність генерувати нові зображення з нуля за допомогою текстової підказки, що описує елементи, які слід включити або виключити з вихідного зображення. Існуючі зображення можуть бути перероблені моделлю для включення нових елементів, описаних текстовою підказкою, за допомогою механізму дифузії-денойзінгу. Крім того, модель також дозволяє використовувати підказки для часткової зміни існуючих зображень за допомогою інпейнтингу та аутпейнтингу, коли використовується відповідний інтерфейс користувача, який підтримує такі функції.

Різні версії Stable Diffusion включають поліпшення та нові можливості. Наприклад, версія XL 1.0 має 3.5 мільярди параметрів, що робить її приблизно в 3.5 рази більшою, ніж попередні версії. Версія 3.0 внесла повністю нову архітектуру замість U-Net, використовуючи Rectified Flow Transformer, що реалізує метод виправленого потоку з архітектурою Transformer.

Stable Diffusion має певні обмеження та виклики, зокрема питання зменшення якості зображень при відхиленні від стандартної роздільної здатності та складнощі з генерацією людських кінцівок і облич. Також існує проблема доступності для індивідуальних розробників через високі вимоги до обчислювальних ресурсів для додаткового навчання моделі.

На відміну від деяких інших моделей, таких як DALL-E, Stable Diffusion надає свій вихідний код і ваги моделі для загального доступу під ліцензією Creative ML OpenRAIL-M. Це дозволяє користувачам вільно використовувати модель для комерційних та некомерційних цілей, з деякими обмеженнями на використання для шкідливих або незаконних цілей. Stable Diffusion представляє

собою значний крок вперед у генерації зображень за допомогою штучного інтелекту, пропонуючи потужні інструменти для створення високоякісних візуальних матеріалів. Водночас модель стикається з технічними, етичними та правовими викликами, які вимагають уваги та відповідального підходу до використання та подальшого розвитку технології. Незважаючи на ці виклики, відкритий характер моделі та її доступність для широкого кола користувачів роблять її цінним інструментом для досліджень, інновацій та творчості в різних областях, від мистецтва до технічних дисциплін. Зростання популярності Stable Diffusion та її використання в різних проєктах свідчить про великий потенціал моделі у сфері генеративного інтелекту. Розвиток та вдосконалення моделі продовжуються, пропонуючи нові можливості для створення якісного візуального контенту та розширення меж можливостей штучного інтелекту в генерації зображень.

1.5 Висновки до розділу 1

Проведений аналіз показав, що генеративний інтелект (ГІ) є ключовим напрямком сучасних досліджень у галузі штучного інтелекту та комп'ютерного зору. Використання генеративного інтелекту для генерування біомедичних зображень є перспективним і необхідним напрямком, який дозволяє синтезувати необхідні набори даних для навчання глибоких нейронних мереж. У розділі проаналізовано гістологічні зображення, які будуть використання для генерування штучних зображень. Аналіз підходів до генерування біомедичних зображень показав, що ефективним підходом є підхід на основі дифузійних моделей. Виділено основні критерії порівняння програмних засобів генерування зображень і показано, що програмний засіб Stable Diffusion є ефективним для генерування біомедичних зображень.

2 АЛГОРИТМИ СИНТЕЗУ ЗОБРАЖЕНЬ НА ОСНОВІ ГЕНЕРАТИВНОГО ІНТЕЛЕКТУ

2.1 Поняття дифузії та дифузійних процесів

Приведемо означення дифузії.

Дифузія – це процес в якому частинки (молекули атоми) перемішуються із області з високою концентрацією у область з меншою концентрацією до вирівнювання концентрації у просторі.

Дифузії бувають таких типів:

1. Молекулярна
2. Теплова
3. Електро-дифузія

Дифузія в природі спостерігається у фізиці, біології, хімії, технологіях, тощо.

Математично процес дифузії описується рівнянням Фіка:

$$\frac{\partial C}{\partial T} = D\nabla^2 C,$$

де C – концентрація речовини;

D – коефіцієнт дифузії;

∇^2 – оператор Лапласа який описує зміну концентрації у просторі.

У інформаційних технологіях (ШІ) використовують дифузійні моделі.

Отже, дифузійні моделі штучного інтелекту – це сукупність генеративних моделей які додають або видаляють шум до вхідних даних.

Ці моделі використовують для генерування нових даних (зображення, аудіо, текст).

Генерування даних складається з двох процесів:

1. Прямий процес – початкові дані поступово зашумлюються аж до перетворення їх у випадковий шум. При цьому використовують гаусівський шум, та множину коефіцієнтів які регулюють рівні шуму на кожному кроці.

2. Зворотний процес – очищення зашумлених даних до відновлення початкової форми. Для цього використовують параметризовану модель що передбачає розподіл даних на кожному кроці.

Для опису дифузійних процесів використовують такі математичні структури [12]:

1. Ланцюги Маркова
2. Гаусівські процеси
3. Нейронні мережі
4. Оптимізаційні методи

На рисунку 2.1 показані математичні структури для опису дифузійних процесів.

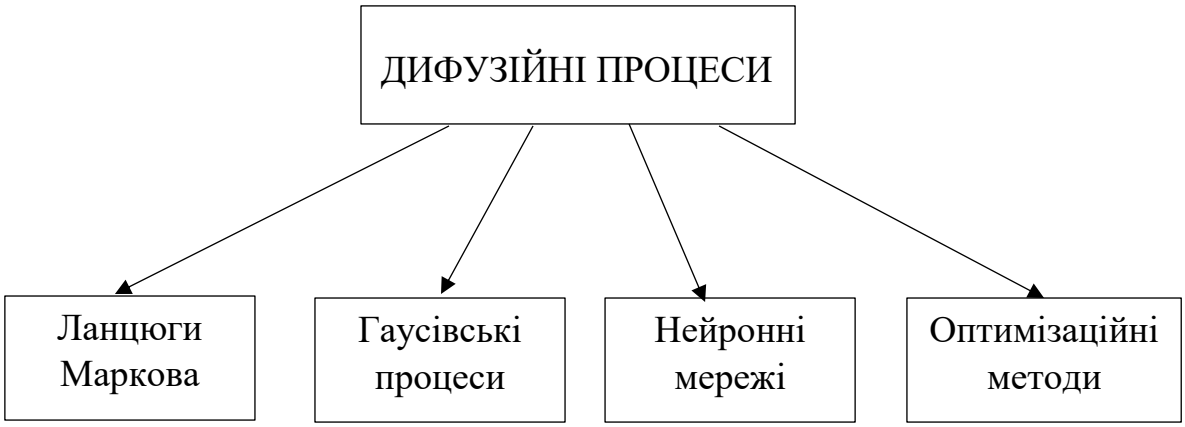


Рисунок 2.1 – Математичні структури для опису дифузійних процесів

2.2 Поняття ланцюгів Маркова і гаусівських процесів

Введемо поняття випадкового процесу.

Означення. Випадковим процесом називається множина випадкових величин, залежних від одного або декількох змінних параметрів. Випадковий процес описується випадковою функцією $X(t), t \in T$. Випадкова функція може бути неперервною або дискретною. Випадкова функція з дискретним часом коли система може змінюватися в моменти часу $t_1, t_2 ... t_n$ дискретної множини часу. Випадкова функція з неперервним часом якщо система може переходити із стану в стан в будь який момент часу t [12]. Випадковий процес $X(t)$ називається марковським, або процесом без післядії, якщо імовірність будь якої події, пов'язаної з протіканням процесу в майбутньому не змінюється якщо ми знаєм теперішній стан.

Випадковий процес називається марковським якщо для всіх t_1, t_2 з області визначення T таких що $t_1 < t_2$ і всіх заданих значеннях $X(t)$ цього процесу $t \le t_1$ умовний розподіл ймовірності $X(t_2)$ залежить лише $X(t_1)$. Для випадку n-вимірної густини ймовірності марковського випадкового процесу виражається через одновимірну та відповідні умовні густини так:

$$p(x_1,...,x_{n-1};t_1,...t_{n-1}) = p_1(x_1,t_1) \cdot p_2(x_2,t_2)|(x_1,t_1) \cdot ... p_n(x_n,t_n \mid x_{n-1},t_{n-1})$$

Марковський процес у якому всі перетини є дискретними випадковими змінними зі скінченною множиною станів називають ланцюгом Маркова. Поділяють ланцюги Маркова з дискретним $(i = 0,1,2...)$ і неперервним часом $0 \le t \le T$.

Опис та зображення ланцюгів Маркова. Нехай заданий випадковий процес $X(t)$ з дискретним часом і дискретними випадковими змінними X_i де

$X_i = X(t_i),\ i = 0,1,2...k$ з однією і тією ж областю значень та різними розподілами імовірностей

Таблиця 2.1 – Розподіл імовірностей для заданого випадкового процесу

X_i	X_1	X_2	…	X_k	…
$p^{(i)}$	$p_1^{(i)}$	$p_2^{(i)}$	…	$p_k^{(i)}$	…

Областю визначення такого процесу є дискретна множина T. Елемнти цієї множини є $t_0, t_1, t_2...t_k$ задовілняють умові $t_0 < t_1 < t_2... < t_k < ...$. Сукупність значень випадкових змінних є областю значень даного процесу. Випадкові змінні будемо називати станами процесу $E_k(k = 1,2...)$. Перехідну імовірність процесу зі стану i в якому він перебував в момент часу t_n (на n кроці) у стан j в момент часу t_{n+1} на n+1 кроці позначимо:

$$p_{ij}^{n,n+1} = p\left(X_{n+1} = j \mid X_n = i\right)$$

$$\sum_{j=1} p_{ij}^{n,n+1} = 1, i = 1,2...k$$

Марковський випадковий процес який заданий на дискретній множині, область значень якого на кожному перетині є скінченою множиною називають ланцюгом Маркова з дискретним часом.

Ми будемо розглядати однорідні ланцюги маркова.

Ймовірність p_{ij} переходу зі стану E_i у стан E_j за один крок (одну позицію часу) об’єднують у матрицю:

$$p = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ p_{k1} & p_{k2} & \cdots & p_{kk} \end{pmatrix}$$

Квадратна матриця з невід'ємними елементами сума яких у кожному рядку =1, називається стохастичною, наприклад:

$$p = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ & & \end{pmatrix}.$$

Коли відбулася складна подія з такими станами $E_{j0}, E_{j1}, ..., E_{jn}$ то це називають конфігурацією. Послідовність спроб називається ланцюгом Маркова якщо імовірність появи конфігурації виду:

$$E_{j0}, E_{j1}, ..., E_{jn} = p\left(E_{j0}, E_{j1}, ..., E_{jn}\right) = p\left(E_{j0}\right) \cdot p\left(E_{j1} \mid E_{j0}\right) ... p\left(E_{jn} \mid E_{jn-1}\right).$$

В початковий момент часу t_0 система перебуває в якомусь стані. У певному стані, система може перебувати лише з деякою імовірністю. Ланцюг Маркова задано якщо відомі імовірності появи події в початковій спробі та умовній імовірності p_{jk} появи події E_k в наступній спробі якщо перед випробовуванням відбувалася подія:

$$a_i\left(a_i = p(E_i)\right) = p\left(X(t_0) = E_i\right),$$

$$E_j\left(p_{jk} = p(E_k \mid E_j) = p(X(t_{n-1}) = E_k \mid X\left(t_n\right) = E_j)\right).$$

Імовірності задамо у вигляді вектора:

$$a = (a_1, a_2, ...); a_i \geq 0; \sum_{i=1} a_1 = 1.$$

Отже ланцюг Маркова заданий тоді й лише тоді, коли відомо, що стохастичний вектор та стохастична матриця однакові розмірності яка є кількістю станів відповідної системи. Вектор A – це вектор імовірностей перебування певної системи в різних станах у початковий момент в стохастична матриця p – це матриця імовірностей переходу системи зі стану в стан за крок. Задання стохастичного вектора та стохастичної матриці ланцюга Маркова називають його матричним зображенням. Для ланцюга Маркова будують граф. Представлення ланцюга Маркова у вигляді графа називають графічним зображенням. На рисунку 2.2 представлено графічне зображення ланцюга Маркова.

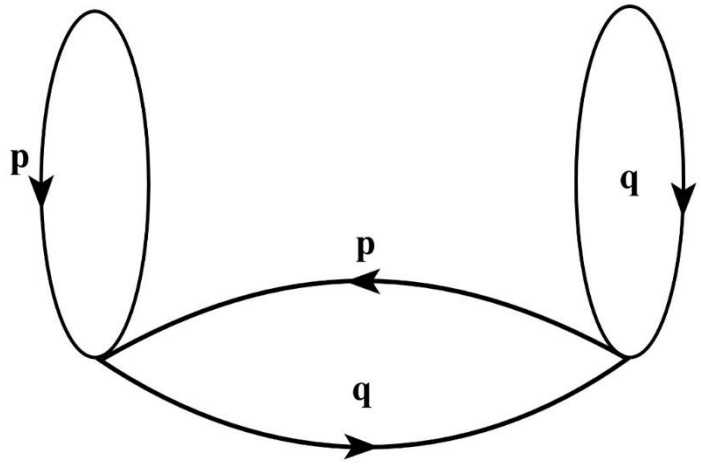


Рисунок 2.2 – Графічне представлення ланцюга Маркова

Матриця для даного ланцюга буде така:

$$P\begin{pmatrix} p & q \\ p & q \end{pmatrix}.$$

Гаусівський процес.

Гаусівський випадковий процес базується на нормальному законі розподілу Гауса [30]. Нормальний закон розподілу є граничним, до якого наближаються інші закони розподілу. Нормальний закон розподілу випадкової величини X має таку густину розподілу:

$$f(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e - \frac{(X-E)^2}{2\sigma^2},$$

де E – математичне сподівання X;

σ – середньоквадратичне відхилення X.

Графічно це виглядає як на рисунку 2.3.

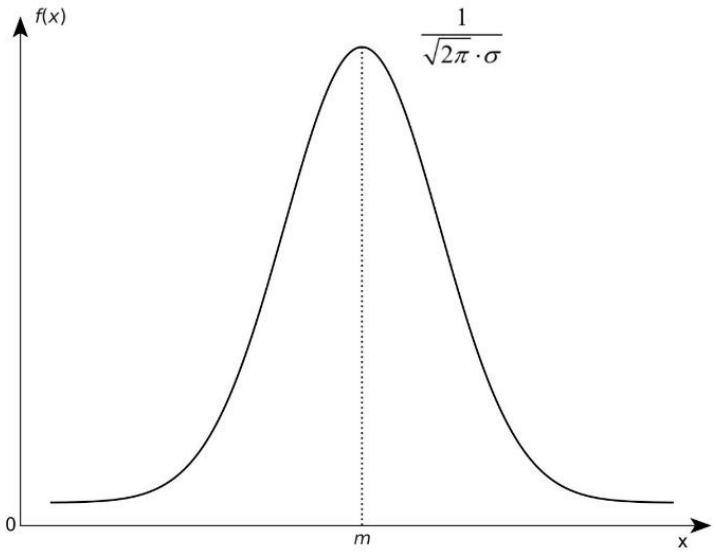


Рисунок 2.3 – Графічне представлення кривої Гауса

Математичне сподівання для нормального закону рівне:

$$E(X) = m.$$

Відповідно дисперсія рівна:

$$D(X) = \sigma^2.$$

Отже, параметр m – це центр симетрії кривої Гауса, тобто він характеризує положення кривої. Параметр σ – характеризує ступінь розсіювання випадкової величини X. На рисунку 2.4 показана залежність кривої Гауса від середньоквадратичного відхилення. Чим більше значення середньоквадратичного відхилення, тим плоскішим буде графік кривої Гауса.

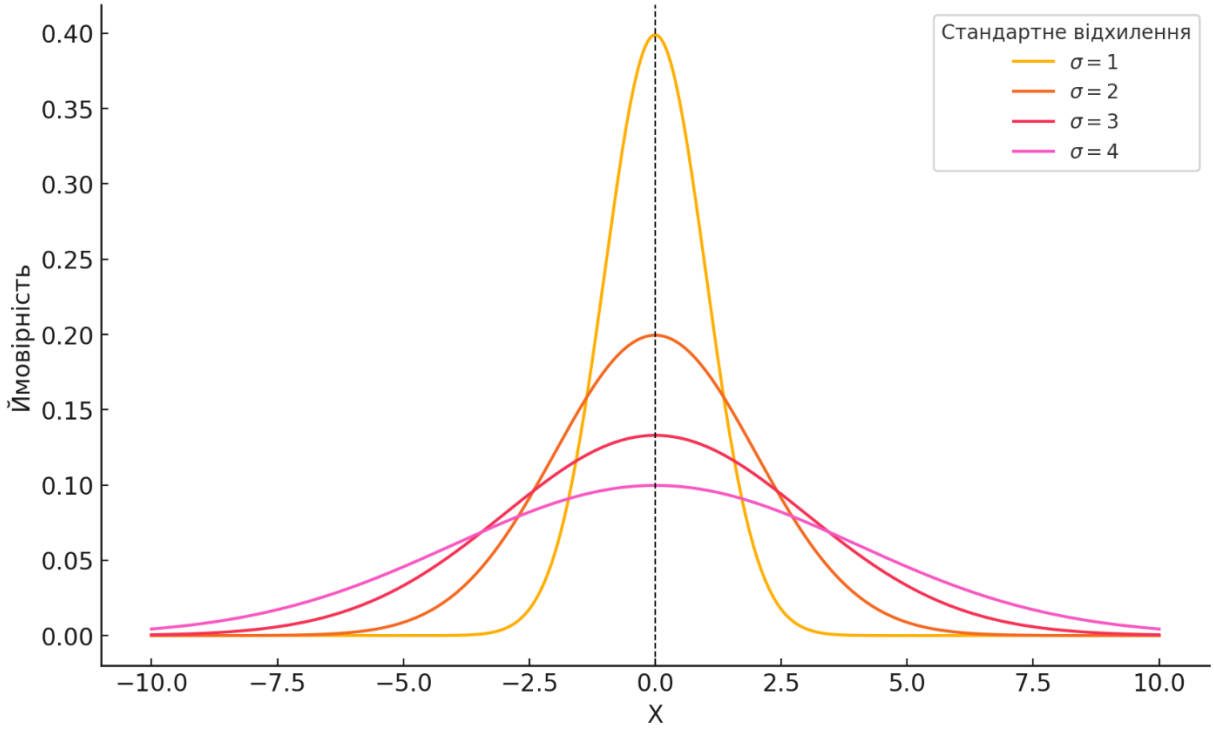


Рисунок 2.4 – Залежність кривої Гауса від середньоквадратичного відхилення

Дисперсія випадкової величини X рівна $D(X) = \sigma^2$.

2.3 Структура U-Net мережі

U-Net – це нейронна мережа, яка вперше була запропонована для задач сегментації медичних зображень, але отримала значно ширше застосування, включаючи Stable Diffusion, де вона є основною частиною для обробки латентних представлень і знешумлення [32-33]. У контексті Stable Diffusion, U-Net використовується як основна частина, яка виконує оцінку шуму $\text{т}_\theta(x_t, t)$ і визначає, як зменшити рівень шуму на кожному кроці зворотного дифузійного процесу.

Розглянемо структуру U-Net. U-Net має симетричну архітектуру, яка складається з двох основних частин:

1. Звужувальний шлях (Downsampling Path). Цей шлях відповідає за витягування характеристик з вхідних даних шляхом послідовних згорток і зменшення розмірності.
2. Розширювальний шлях (Upsampling Path).

Цей шлях відновлює просторову роздільну здатність через транспоновані згортки та інтегрує інформацію високого рівня із шляху зменшення. Між цими двома шляхами є пропускні зв’язки (skip connections), які передають дані з відповідних рівнів звужувального шляху до розширювального. Це допомагає зберігати інформацію про низькорівневі деталі.

Розглянемо детальніше звужувальний шлях.

На кожному рівні l, вхідний тензор x^l проходить через набір згорткових операцій:

$$x^{l+1} = \sigma\left(W^{(l)} * x^l + b^{(l)}\right),$$

де $W^{(l)}$ – ядро згортки;

$b^{(l)}$ – вектор зсуву;

$*$ – операція згортки;

σ – функція активації (наприклад, ReLU).

Зменшення розмірності досягається через пулінг згідно виразу:

$$x^{l+1} = \text{MaxPooling}(x^l).$$

Розширювальний шлях.

Розширення розмірності виконується за допомогою транспонованих згорток згідно виразу:

$$\text{Conv}^T(x) = W^T * x + b,$$

де W^T – ядро транспонованої згортки;

$*$ – транспонована згортка;

b – зміщення.

Після транспонованої згортки додаються пропускні зв’язки:

$$x^{l-1} = x^{l-1} + x^l_{\text{skip}}.$$

Пропускні зв’язки передають інформацію від звужувального шляху до розширювального, що покращує деталізацію:

$$x^l_{\text{skip}} = \text{Concatenate}(x^l_{\text{contract}}, x^l_{\text{expand}})$$

Вхідними даними для U-net мережі є:

1. Латентне представлення x_t.
2. Шум є.
3. Текстова підказка, закодована через CLIP.

4. Часовий маркер t, який додається до даних через синусоїдальні кодування.

Для Stable Diffusion використовують модифіковану U-net.

Пропускні зв'язки інтегрують текстову інформацію через механізми перехресної уваги (cross-attention) у латентному просторі:

$$z' = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

де Q, K, V – матриці запитів, ключів і значень для текстової підказки.

Уточнимо архітектуру U-Net для Stable Diffusion.

Для звужувального шляху використовуються блоки ResNet, які ідентифікують характеристики, а кожному рівні є перехресна увага для інтеграції текстової підказки.

Для звужувального шляху використовуються транспоновані ResNet-блоки та інтегруються пропускні зв'язки з відповідного рівня звужувального шляху.

Для U-Net мережі у Stable Diffusion мінімізується різницю між передбаченим реальним і шумом згідно виразу:

$$L = \text{E}_{x_0,\text{т},t}\left[\left\|\text{т} - \text{т}_\theta(x_t, t)\right\|^2\right],$$

де т – реальний шум;

$\text{т}_\theta(x_t, t)$ – передбачений шум.

На рисунку 2.5 представлена архітектура U-Net, яка використовується в Stable Diffusion. Ця архітектура складається з двох основних частин: звужувальної (downsampling) та розширювальної (upsampling), а також середнього блоку (middle block). Розглянемо детальніше кожен блок.

На вхід звужувального блоку поступає зображення розмірністю 128 × 128, яке, послідовно проходить через кілька шарів , кожен із яких зменшує розмір.

Кожен шар складається із згорток (convolutions), нормалізації та функцій активації.

Середній блок (Middle Block) використовується для обчислення особливостей на найменшій розмірності 4×4 і включає кілька згорткових шарів і можливі механізми уваги (attention), які аналізують взаємозв'язок між особливостями на цьому рівні.

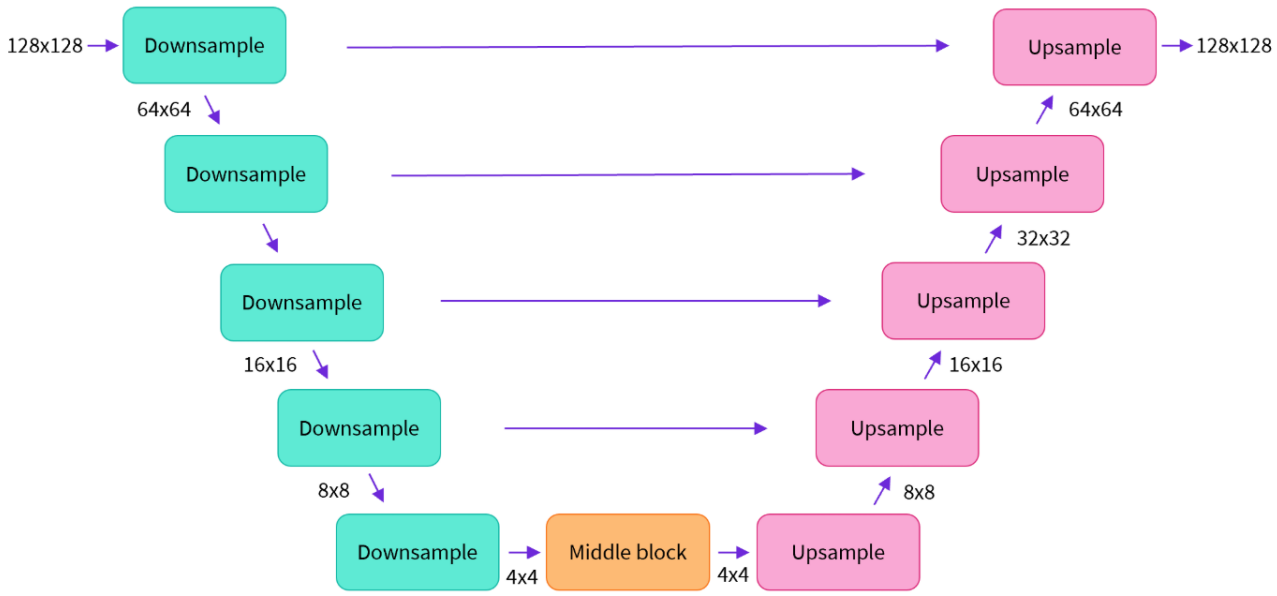


Рисунок 2.5 – Архітектура U-Net в Stable Diffusion

Розширювальний блок (Upsampling) відновлює просторову розмірність тензора, поступово повертаючи розміри до 128×128. Він включає транспоновані згортки (transposed convolutions), збільшуючи розмірність згідно ряду: $4 \times 4 \rightarrow 8 \times 8 \rightarrow 16 \times 16 \rightarrow 32 \times 32 \rightarrow 64 \times 64 \rightarrow 128 \times 128$.

У кожному блоці використовуються пропускні з'єднання (skip connections) із відповідного шару звужувальної блоку, щоб передати локальні особливості.

Формальний опис процесу опрацювання зображення у кожному блоці наступний. У звужувальному блоці опрацювання відбувається згідно виразу:

$$z_{\text{downsample}} = \text{Conv}(\text{Pool}(z_{\text{input}})),$$

де Pool — операція пулінгу, Conv — операція згортки.

У середньому блоці опрацювання відбувається згідно виразу:

$$z_{\text{middle}} = \text{Attention}(\text{Conv}(z_{\text{downsample}})).$$

У розширювальному блоці вираз такий:

$$z_{\text{upsample}} = \text{Conv}^T(z_{\text{middle}}) + z_{\text{skip}},$$

де z_{skip} – пропускне з'єднання.

Вираз для крос-уваги (Cross-Attention) такий:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V,$$

де Q,K,V — запити, ключі та значення для уваги.

Ця архітектура є гнучкою та потужною для генеративних завдань. Вона забезпечує ефективне видалення шуму на кожному етапі, використовуючи локальні особливості та текстові підказки. Завдяки своїй модульній структурі, U-Net у Stable Diffusion здатен працювати з латентним простором, оптимізуючи якість створених зображень.

2.4 Метод градієнтного спуску

Основним завданням в машинному навчанні є зменшення заданої функції помилки.

По суті, це означає, що це є задача оптимізації : по заданій функції знайти аргументи, в яких ця функція максимізується або мінімізується [34-35].

Найчастіше мережі побудовані так, що ваги можуть бути довільними дійсними числами. Це означає, що багато складності, пов'язані з формою допустимих множин, крайовими ефектами та іншими лямбда-множниками Лагранжа. З іншого боку, сам функція, яку ми оптимізуємо, нам зазвичай ні в якому вигляді не задана. Легко оптимізувати ваги одного перцептрону - це опукла функція, яке можна було розв’язати в явному вигляді. Але для випадку оптимізації великої нейронної мережі цільова функція - це дуже складна композиція інших функцій.

Найпростіший приклад опуклою функції - це параболоїд, функція другого порядку від своїх аргументів. Прикладом є лінійна регресія і функція помилки одного перцептрону:

$$E(w_1,...,w_n) = -\frac{1}{2}\sum_{i=1}^{D}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 .$$

Екстремум функції буде один, а тому автоматично глобальним.

Але у багатошарових нейронних мережах функція помилки має складний характер з великим числом локальних максимумів і мінімумів. Формально це означає, що похідна (градієнт) стає нулем багато разів в різних точках.

Зауважимо ще, що різних екстремумів може бути багато. Кількість екстремумів росте експоненціально від числа нейронів (тобто від числа аргументів функції, яку ми оптимізуємо).

Методи оптимізації в основному займаються тим, що прискорюють градієнтний спуск, розробляють різні його варіанти, додають обмеження, але принципово інших методів не існує. Тому для використання градієнтного спуску можна використати такими прийомами:

1) якщо похідна є, але обчислити її неможливо, то можна обчислити тільки значення функцій в різних точках, а похідну можна підрахувати приблизно; наприклад, прямо за визначенням:

$$f'(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon},$$

тобто, якщо взяти досить малий ε , похідну можна наближено підрахувати більш точними формули кінцевих різниць, наприклад:

$$f'(x) \approx \frac{f(x-2\epsilon) - 8f(x-\epsilon) + 8f(x+\epsilon) - f(x+2\epsilon)}{12\epsilon}.$$

2) якщо похідної немає, то наближати функцію можна за допомогою функції, яка подібна на функцію, що оптимізується;

3) якщо похідна очевидна, то оптимізовують не саму функцію якості, спеціальну гладку функцію помилки, яка буде тим більше, ніж «більш помилково» виявиться обчислення релевантності.

Отже, при градієнтному спуску найбільше змінюється той параметр, який має найбільшу за модулем похідну.

Параметр градієнтного спуску – це його *швидкість навчання* η; вона регулює розмір кроку, який ми робимо в напрямку схилу градієнта. У чистому градієнтному спуску швидкість навчання задається вручну, і вона може досить сильно вплинути на результат. Тут можуть виникнути дві протилежні одна одній проблеми:

– якщо кроки будуть занадто маленькими, то навчання буде занадто довгим, і підвищується ймовірність зупинитися на локальному мінімумі;

– якщо кроки є занадто великими, можна нескінченно проходити шуканий мінімум взад-вперед, але так і не прийти в саму нижню точку.

Припустимо, що у нас є набір даних *D*, що складається з пар (*x*, *y*) , де *x* – ознаки, а *y* – правильна відповідь. Модель з вагами θ на цих даних робить деякі передбачення $f(x,\theta)$, $x \in D$, і задана функція помилки *E*, яку можна підрахувати на кожному прикладі, $E(f(x,\theta), y)$. Наприклад, це може бути квадрат або модуль відхилення $f(x,\theta)$ від *y* в разі регресії або перехресна ентропія в разі

класифікації. Загальна функція помилки буде сумою помилок на кожному тренувальному прикладі:

$$E(\theta) = \sum_{(x,y)\in D} E(f(x,\theta), y).$$

А градієнтний спуск буде виглядати так:

$$\theta_t = \theta_{t-1} - \eta \nabla E(\theta_{t-1}) = \theta_{t-1} - \eta \sum_{(x,y)\in D} \nabla E(f(x,\theta_{t-1}), y).$$

Для того щоб зробити один-єдиний крок градієнтного спуску, потрібно пройтися по всій тренувальній множині. А це може бути в реальних задачах дуже великим.

На практиці працює не простий, а стохастичний градієнтний спуск, в якому помилка підраховується і ваги підправляються не після проходу по всьому тренувальної множини, а після кожного прикладу:

$$\theta_t = \theta_{t-1} - \eta \nabla E(f(x_t,\theta_{t-1}), y_t).$$

Основна перевага стохастичного градієнтного спуску полягає в тому, що помилка на кожному кроці обчислюється швидко, ваги змінюються відразу ж, що дуже сильно прискорює навчання. На практиці часто трапляються ситуації, коли навчання фактично вже зійшлося ще до того, як ми навіть один раз пройшли по всіх тренувальним прикладах. Але крім обчислювальних, є і змістовні переваги: стохастичний градієнтний спуск працює «більш випадково», ніж звичайний, і тому можна сподіватися, що він не зупиниться в маленьких локальних мінімумах. Взагалі, в локальної оптимізації часто допомагає намагатися зробити оновлення «якомога більше випадковими», намагатися якомога більше досліджувати в просторі пошуку.

Тому на практиці зазвичай використовується щось середнє: стохастичний градієнтний спускається по міні-батч (mini-batch), невеликим підмножини тренувального набору [36-39]. Це дозволяє зберегти всі плюси стохастичного градієнта (кілька десятків або сотень прикладів - це зазвичай дуже мала частина тренувальної множини), але при цьому користуватися процедурами матричної арифметики, які дуже швидко обчислюються на відеокарті.

2.5 Узагальнений алгоритм синтезу зображень на основі дифузійної моделі

Генерування зображень на основі дифузійної моделі відбувається в програмному середовищі Stable Diffusion. Базова модель Stable Diffusion навчається на великій вибірці зображень [40-42]. Навчання на основі своєї вибірки відбувається в середовищі нейромережі Hypernetwork. Ця мережа корегує ваги базової моделі. Алгоритм генерування зображень на основі дифузійної моделі складається з таких кроків:

1. Навчання на основі свого dataset зображень в середовищі Hypernetwork;
2. Процес зашумлення початкового dataset I_C;
3. Процес знешумлення.

Деталізуємо ці кроки. Початкова вибірка перетворюється в латентний простір: $I_C \rightarrow Z_{0C}$. На основі Z_{0C} обчислюємо значення зашумлення на кожному кроці t так:

$$Z_t = \sqrt{\alpha_t} Z_{0C} + \sqrt{1-\alpha_t}\varepsilon_t,$$

де α_t – коефіцієнт, який визначає швидкість зашумлення на кроці t.

Значення кроку t вибираємо із діапазону:

$$t \in [0, T],$$

де T – кількість кроків;

ε_t – значення випадкового гаусівського шуму на кроці *t*.

Значення ε_t обчислюється згідно виразу:

$$\varepsilon_t \sim N(E, D),$$

де N – нормальний закон розподілу із математичним сподіванням $E = 0$ і дисперсією $D = 1$.

Значення знешумлення обчислюємо згідно виразу:

$$Z_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(Z_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\varepsilon}_t \right),$$

де $\hat{\varepsilon}_t$ – оціночне значення шуму на кроці *t*;

$\bar{\alpha}_t$ – коефіцієнт, який визначає рівень шуму на попередньому кроці *t*;

β_t – коефіцієнт, що контролює рівень зменшення шуму.

Після виконання процесу знешумлення (після проходження $t = T$ кроків) утворюється вектор Z_{1C} у латентному просторі. Потім енкодер перетворює Z_{1C} у множину зображень I_{CD}, причому $I_{CD} \sim I_C$. Якість згенерованих зображень перевіряємо метриками *IS* і *FID*.

Метрика *IS* базується на моделі [43] нейронної мережі для класифікації кольорових зображень Google Inception V3. Ця метрика перевірена на датасеті ImageNet потужністю 1,2 мільйона RGB зображень, які поділені на 1000 класів.

Аналітичний вираз для метрики такий:

$$IS(G) \approx \exp\left(E_{x \sim p_g}\left[D_{KL}\left(p(y \mid x) \| \ p(y)\right)\right]\right)$$

де E – математичне сподівання;

$x \sim p_g$ – показує, що x є зображенням, що синтезоване із розподілу p_g (розподіл генератора);

D_{KL} – є відстанню Кульбака-Лейблера між розподілом умовної імовірності $p(y \mid x)$ та маргінальним розподілом $p(y)$.

Метрика IS вимірює середню відстань Кульбака-Лейблера між умовним розподілом $p(y \mid x)$ та маргінальним розподілом класів $p(y)$. Мінімальним значенням метрики є 1, а максимальним — кількість класів.

Метрика *FID* порівнює розподіли оригінальних та синтетичних даних. На основі цієї метрики відстань між зображеннями обчислюється так:

$$d^2\left((m_r C_r),(m_g C_g)\right) = \| \ m_r - m_g \ \|^2 + \mathrm{Tr}\left(C_r + C_g - 2\left(C_r C_g\right)^{\frac{1}{2}}\right),$$

де $(m_r C_r)$ та $(m_g C_g)$ – середнє та covariance реального та синтезованого розподілу даних відповідно,

Tr – сума діагональних елементів матриці.

Отже, чим менше значення метрики, тим менша відстань між розподілами, тобто зображення більше подібні між собою. Метрика FID чутлива до спотворень на зображеннях (зсув, шум і т.д).

2.6 Висновки до розділу 2

У другому розділі досліджені основні математичні структури, які використовуються для аналізу дифузійних процесів: ланцюги Маркова, гаусівські процеси, глибокі нейронні мережі типу U-net, метод градієнтного спуску. На основі цих математичних структур розроблено узагальнений алгоритм синтезу гістологічних зображень на основі дифузійних моделей.

3 СИНТЕЗ І ТЕСТУВАННЯ ГІСТОЛОГІЧНИХ ЗОБРАЖЕНЬ В СЕРЕДОВИЩІ STABLE DIFFUSION

3.1 Програмне середовище Stable Diffusion

У даному підрозділі опишемо веб-інтерфейс для роботи з моделями Stable Diffusion. AUTOMATIC1111 — веб-інтерфейс для роботи з моделями Stable Diffusion для зручності у використанні генеративних моделей зображень (рисунок 3.1). Він пропонує багато функцій для генерації, редагування та навчання моделей із простим доступом через веб-браузер. AUTOMATIC1111 став основним інструментом для багатьох користувачів завдяки своїй гнучкості та широкому функціоналу.

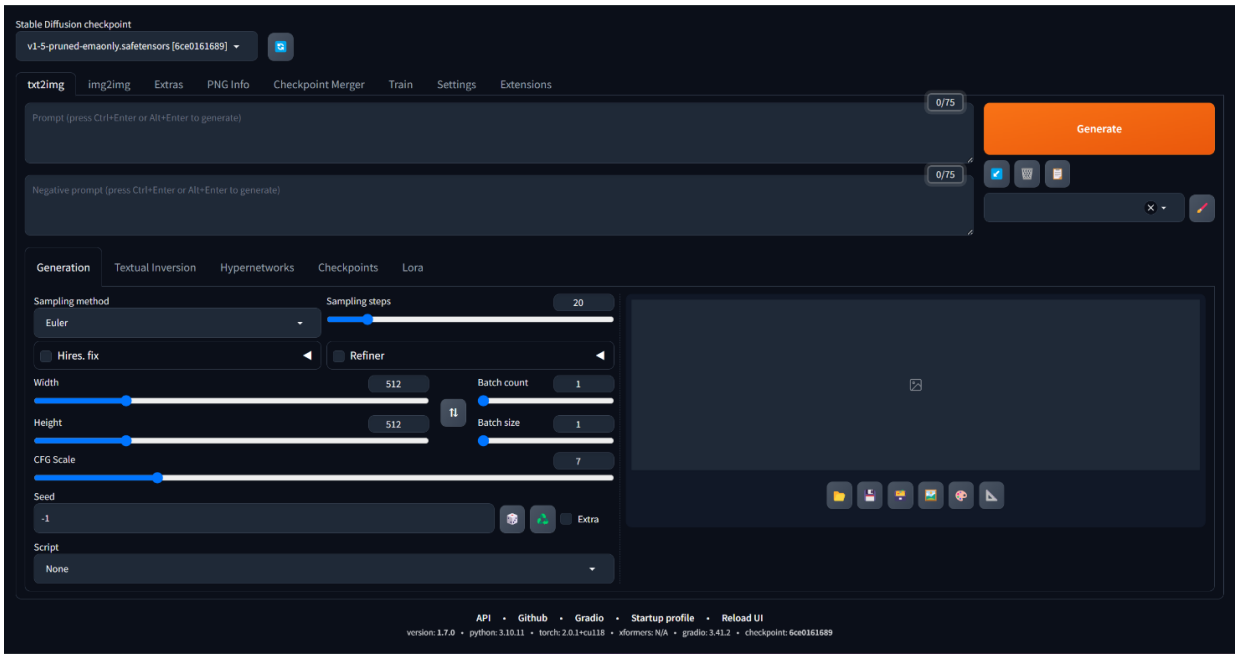


Рисунок 3.1 – Зображення веб-інтерфейсу AUTOMATIC1111 (вкладка Generation)

Він використовується для генерації зображень за текстовими підказками (prompts), а також для різних операцій з моделлю Stable Diffusion. На рисунку 3.2 показано граф інтерфейсу на якому ми можемо бачити його структуру. Нижче приведемо детальний опис інтерфейсу.

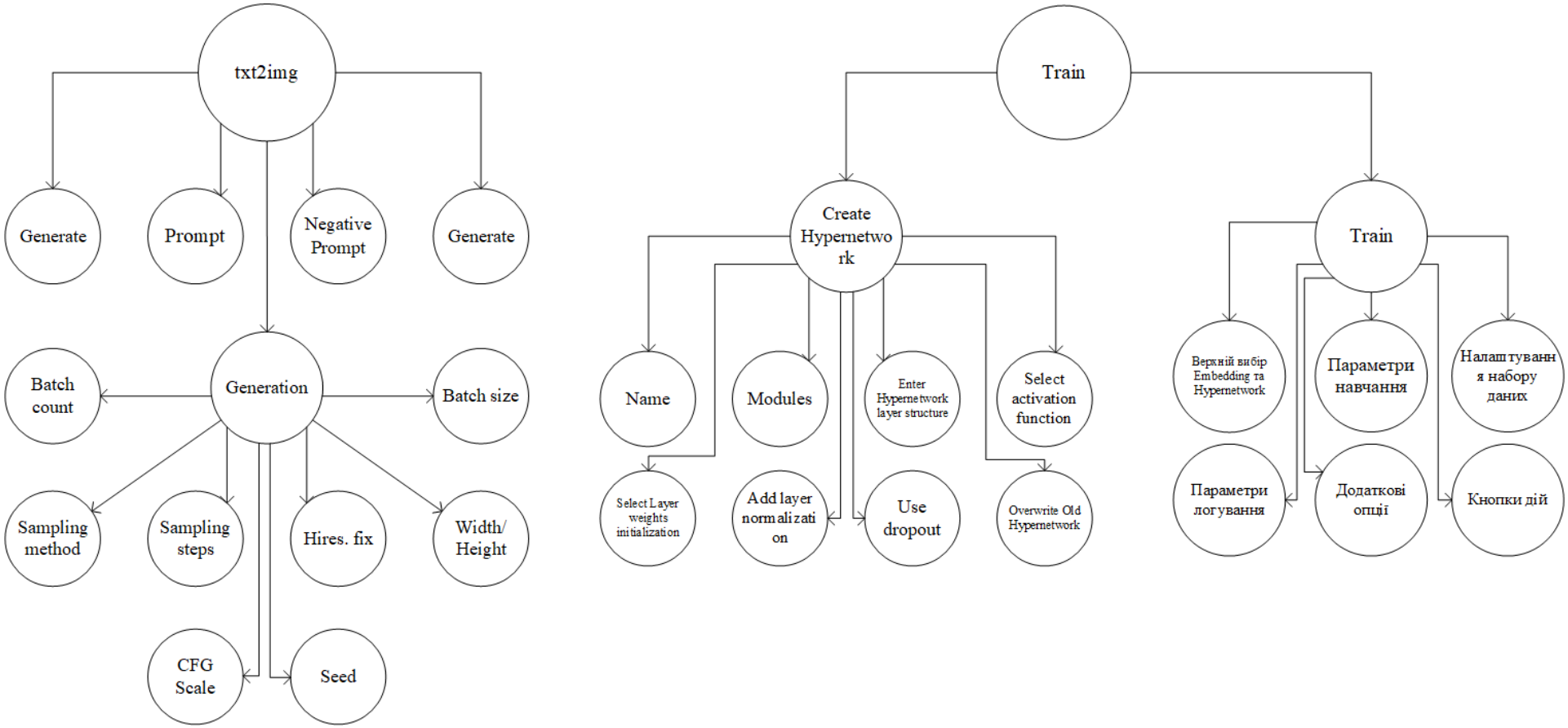


Рисунок 3.2 – Граф інтерфейсу

1. Верхня панель

Checkpoint (v1-5-pruned-emaonly.safetensors): Вибір контрольної точки (моделі), яка буде використовуватися для генерації. У цьому випадку вибрано модель v1-5-pruned-emaonly.safetensors.

Tabs (txt2img, img2img, Extras, PNG Info, Checkpoint Merger, Train, Settings, Extensions):

– txt2img: Генерація зображень на основі текстових підказок.

– img2img: Генерація зображень на основі існуючого зображення із використанням текстових інструкцій.

– Extras: Інструменти для зміни розміру зображень або покращення якості.

– PNG Info: Перегляд метаданих зображення, згенерованого моделлю.

– Checkpoint Merger: Злиття різних контрольних точок.

– Train: Розділ для навчання моделей.

– Settings: Налаштування інтерфейсу та моделі.

– Extensions: Розширення для додаткових функцій.

2. Поля вводу текстових підказок

– Prompt: Сюди вводиться основна текстова підказка, яка визначає, що саме потрібно згенерувати. Ліміт — 75 символів (за замовчуванням).

– Negative Prompt: Вказується, чого не повинно бути в зображенні (наприклад, певних кольорів чи деталей).

3. Панель генерації (Generation Panel)

– Sampling method: Метод вибірки (sampling). Наприклад, у цьому випадку обрано метод Euler. Інші популярні методи: DPM++, LMS, DDIM, тощо. Вибір методу впливає на якість і швидкість генерації.

– Sampling steps: Кількість кроків вибірки. У цьому прикладі обрано 20 кроків, що впливає на деталізацію зображення — більше кроків дає вищу якість, але сповільнює процес.

– Hires. fix: Опція покращення роздільної здатності після генерації зображення. Зазвичай використовується для збільшення чіткості.

– Width/Height: Ширина і висота зображення, яке буде згенеровано. Тут задано 512×512.

– CFG Scale: Параметр керування масштабом умовної ймовірності. У цьому випадку обрано значення 7. Вищі значення роблять зображення більш "схожим" на підказку, але можуть втрачатися деталі.

– Seed: Початкове значення випадкових чисел, що визначає унікальність зображення. Значення -1 означає, що seed буде генеруватися випадково.

4. Параметри обробки

– Batch count: Кількість зображень, які генеруються за один прохід.

– Batch size: Кількість зображень, що генеруються одночасно в одному "батчі".

5. Додаткові кнопки

– Generate (помаранчева кнопка): Запускає процес генерації.

– Кнопки управління (папка, смітник, збереження тощо): Використовуються для завантаження, збереження чи очищення полів підказок.

6. Нижня панель

– API: Вказує, що інтерфейс може використовувати API для інтеграції.

– GitHub: Посилання на репозиторій AUTOMATIC1111.

– Gradio: Використання Gradio для роботи з інтерфейсом.

– Startup profile: Налаштування профілю запуску.

– Reload UI: Оновлення інтерфейсу.

Цей інтерфейс є дуже гнучким і підтримує численні налаштування, що дозволяє користувачам генерувати високоякісні зображення, експериментувати з параметрами вибірки, масштабуванням та іншими особливостями моделі.

Приведемо опис основних параметрів, які вказані в інтерфейсі Automatic1111 stable diffusion webui, та їхнє значення.

1. Sampling Method (Метод вибірки)

– Визначає алгоритм, який використовується для ітерацій у процесі дифузії.

Приклади:

– Euler: Простий і швидкий метод вибірки, добре підходить для невеликих зображень або швидкої генерації.

– Euler a: Анімаційний варіант Euler із більшою варіативністю.

– DPM++ 2M: Покращений метод, добре підходить для більш точного контролю якості.

– DDIM: Детальний метод, який зазвичай використовується для генерації високоякісних зображень.

Вплив: Різні методи вибірки по-різному впливають на стиль, якість і швидкість генерації.

2. Sampling Steps (Кроки вибірки)

– Визначає кількість ітерацій, які буде виконано під час дифузії.

– Типові значення: Від 20 до 50.

Вплив:

– Менша кількість кроків (наприклад, 10–20): Швидше виконання, але можливе зниження деталізації.

– Більша кількість кроків (50+): Більша деталізація, але довший час генерації.

3. Width / Height (Ширина / Висота)

– Розмір зображення, що буде згенеровано, у пікселях.

– Типові значення: 512×512, 768×768 тощо.

Вплив:

– Збільшення розмірів потребує більше обчислювальних ресурсів і пам'яті GPU.

– Для великих розмірів можна використовувати опцію "Hires. fix" для оптимізації.

4. CFG Scale (Classifier-Free Guidance Scale)

– Параметр, який контролює, наскільки строго зображення повинно відповідати текстовій підказці.

– Типові значення: 5–15.

Вплив:

– Низьке значення (<7): Генерує більш "вільні" варіації зображення.

– Високе значення (>10): Зображення буде строго відповідати текстовій підказці, але може втрачати художню варіативність.

5. Seed (Початок генерації)

Числове значення, яке визначає початковий стан випадкових чисел для генерації.

Особливості:

– -1: Генерує випадкове значення seed при кожній генерації.

– Конкретне число (наприклад, 42): Забезпечує відтворюваність зображень для однакових параметрів.

Вплив:

Змінюючи seed, можна отримати різні результати при тій самій текстовій підказці.

6. Batch Count (Кількість партій)

– Визначає, скільки партій зображень потрібно згенерувати за один запуск.

– Типові значення: 1–5.

Вплив:

Кількість партій визначає загальну кількість зображень, які будуть згенеровані в один прохід.

7. Batch Size (Розмір партії)

– Кількість зображень, що генерується одночасно в одній партії.

– Типові значення: 1–4.

Вплив:

Визначає, скільки зображень буде генеруватися паралельно. Вищі значення потребують більше пам'яті GPU.

8. Hires. Fix (Високоякісна обробка)

– Опція для покращення деталізації зображення після його первинної генерації.

Особливості:

– Використовується для збільшення роздільної здатності.

– Застосовує додаткові кроки апсемплінгу.

Вплив:

Покращує чіткість великих зображень, але значно збільшує час генерації.

9. Negative Prompt (Негативна підказка)

– Список об'єктів, стилів або деталей, яких потрібно уникати в зображенні.

Наприклад:

– "grainy, blurry, low quality" для уникнення зернистості та розмитості.

Вплив:

Допомагає очистити зображення від небажаних елементів.

Ці параметри надають великий контроль над процесом генерації зображень і дозволяють отримати точні результати, які відповідають вашим цілям. Різні комбінації цих параметрів можуть суттєво впливати на стиль, якість і швидкість генерації.

На рисунку 3.3 показано інтерфейс вкладки Train, Create Hypernetwork у Stable Diffusion WebUI від AUTOMATIC1111. У цій секції користувач може створювати власні гіпермоделі (Hypernetwork), які дозволяють адаптувати генерацію під конкретні стилі, концепції або набори даних.

Приведемо детальний опис параметрів вкладки.

Основні параметри для створення гіпермоделі (Hypernetwork):

1. Name (Назва): Поле для введення назви гіпермоделі, яку ви створюєте. Назва допомагає ідентифікувати гіпермодель для подальшого використання.

2. Modules: Модулі (вибір рівнів роздільної здатності), які будуть використовуватися в гіпермоделі.

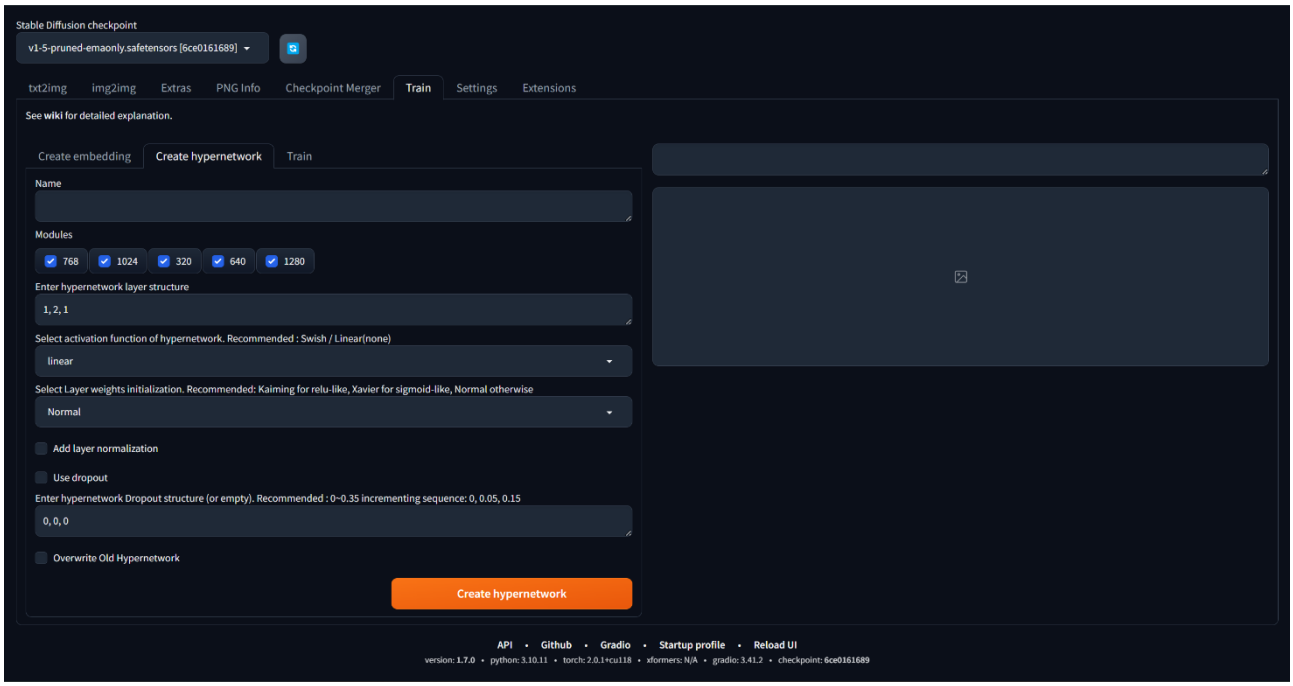


Рисунок 3.3 – Зображення веб-інтерфейсу AUTOMATIC1111

– Доступні значення: 768, 1024, 320, 640, 1280.

– Вибір залежить від того, якого рівня деталізації хочемо досягти в навчанні.

3. Enter hypernetwork layer structure: Визначає структуру шарів у гіпермоделі. Наприклад, 1, 2, 1 означає, що буде використано три шари з відповідною кількістю вузлів (нейронів).

4. Select activation function of hypernetwork: Вибір функції активації для гіпермоделі. Рекомендовані значення:

– Swish: Нелінійна функція активації, яка добре підходить для багатьох генеративних завдань.

– Linear (none): Лінійна функція без додаткової активації.

5. Select Layer weights initialization: Спосіб ініціалізації ваг шарів нейронної мережі.

Рекомендації:

– Kaiming: Підходить для активацій на основі ReLU.

– Xavier: Для активацій sigmoid-подібних функцій.

– Normal: Використовується в загальних випадках.

6. Add layer normalization:

– Опція для додавання нормалізації шарів (Layer Normalization). Використовується для стабілізації навчання.

7. Use dropout:

– Додавання Dropout у шари гіпермоделі. Dropout допомагає уникати перенавчання.

– В полі Enter hypernetwork Dropout structure можна вказати значення Dropout для кожного шару у форматі: 0, 0.05, 0.15.

8. Overwrite Old Hypernetwork:

– Якщо увімкнено, дозволяє перезаписати існуючу гіпермодель із такою ж назвою.

Дії з гіпермоделлю:

1. Create hypernetwork: Помаранчева кнопка запускає процес створення гіпермоделі з вказаними параметрами.

Гіпермоделі дозволяють налаштувати модель Stable Diffusion для генерації зображень зі стилями або характеристиками, які відповідають вашим даним.

Ця секція інтерфейсу надає велику гнучкість у налаштуванні гіпермоделей, що робить їх корисним інструментом для персоналізації та покращення результатів генерації.

На рисунку 3.4 показана вкладка Train у Stable Diffusion WebUI, яка використовується для навчання гіпермоделей (Hypernetwork) або ембеддингів (Embedding). У цьому інтерфейсі можна налаштовувати параметри навчання та запускати процеси навчання моделей на власних наборах даних.

Приведемо детальний опис параметрів вкладки.

1. Верхній вибір Embedding та Hypernetwork

– Вибір, що саме тренувати: ембеддинг (Embedding) або гіпермодель (Hypernetwork).

2. Параметри навчання:

2.1 Learning Rate (Embedding та Hypernetwork):

– Embedding Learning Rate: Швидкість навчання для ембеддингів. Наприклад, тут значення 0.005.

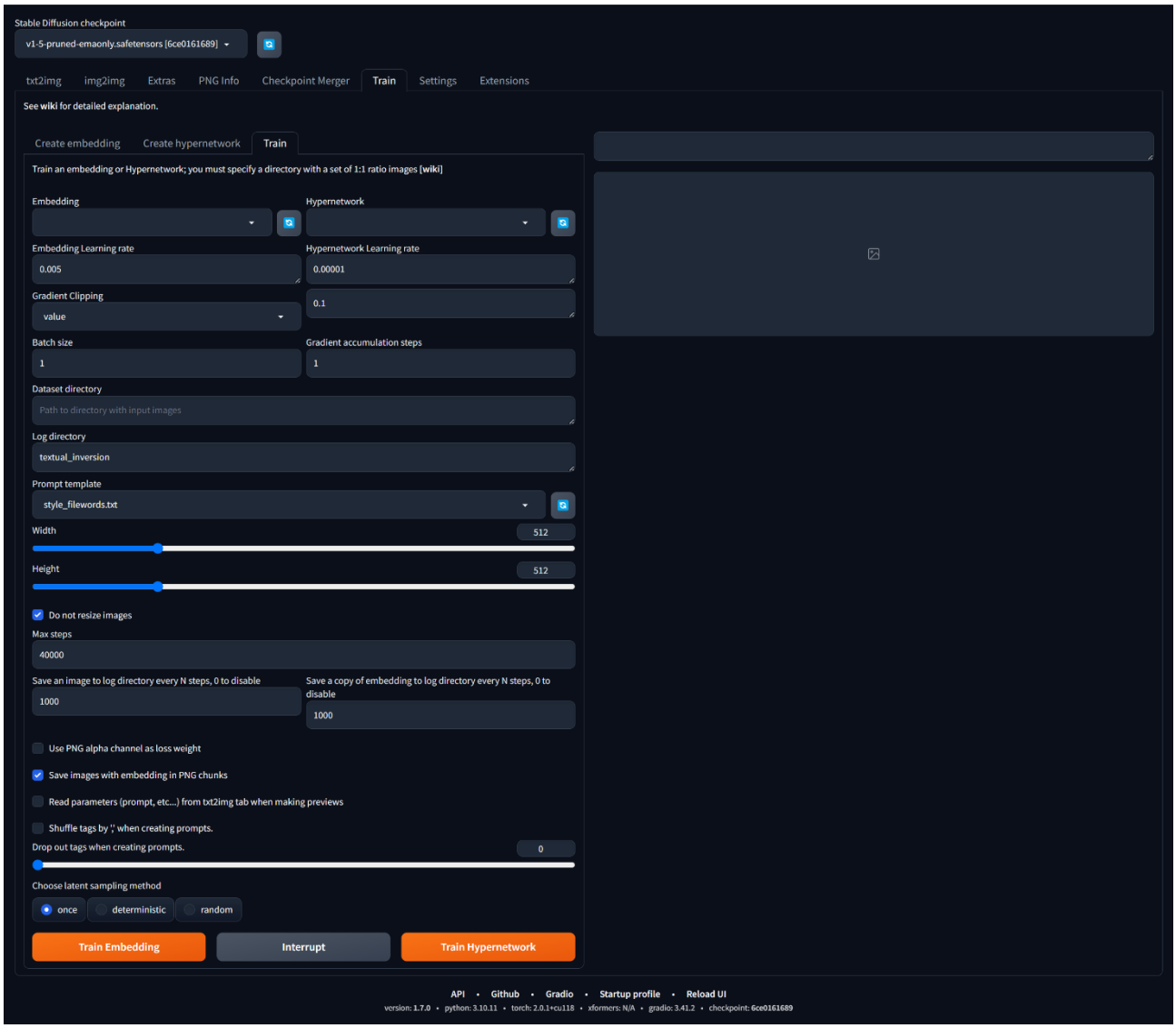


Рисунок 3.4 – Зображення веб-інтерфейсу AUTOMATIC1111 (вкладка Train)

– Hypernetwork Learning Rate: Швидкість навчання для гіпермоделей (наприклад, 0.00001). Важливо для тонкого налаштування навчання.

2.2 Gradient Clipping:

– Параметр, який обмежує величину градієнта, щоб уникнути нестабільності під час навчання.

– Value: Максимальне значення градієнта (наприклад, 0.1).

2.3 Batch Size:

– Кількість зображень, які обробляються в одному батчі.

– Наприклад, значення 1 означає, що модель обробляє одне зображення за один раз.

2.4 Gradient Accumulation Steps:

– Кількість кроків, за які накопичуються градієнти перед оновленням ваг.

3. Налаштування набору даних

3.1 Dataset Directory:

– Шлях до папки з вхідними зображеннями, які будуть використовуватися для навчання.

3.2 Log Directory:

– Каталог для збереження проміжних результатів або логів.

3.3 Prompt Template:

– Шаблон для генерації текстових підказок під час навчання. У цьому прикладі використовується файл style_fireworks.txt.

3.4 Width та Height:

– Роздільна здатність вхідних зображень. У цьому прикладі встановлено 512×512512 \times 512512×512.

3.5 Do not resize images:

– Якщо активовано, зображення не будуть змінювати розмір під час навчання.

4. Параметри логування

4.1 Max Steps:

– Максимальна кількість ітерацій навчання. У цьому прикладі це 40,000 кроків.

4.2 Save an image to log directory every N steps:

– Інтервал, через який модель зберігає результати навчання у лог.

5. Додаткові опції

5.1 Use PNG alpha channel as loss weight:

– Використовує альфа-канал PNG як вагу для втрат.

5.2 Shuffle tags by ';' when creating prompts:

– Перемішує теги підказок, розділених символом.

5.3 Choose latent sampling method:

– Вибір методу вибірки для латентного простору:

– once: Одноразова вибірка.

– deterministic: Детермінована вибірка.

– random: Випадкова вибірка.

6. Кнопки дій

– Train Embedding: Запуск навчання ембеддингу з вказаними параметрами.

– Train Hypernetwork: Запуск навчання гіпермоделі з вказаними параметрами.

– Interrupt: Зупинка процесу навчання.

Цей інтерфейс дозволяє гнучко налаштовувати параметри навчання для адаптації моделі під конкретні задачі, наприклад, створення стилізованих зображень або додавання нових концепцій до моделі.

3.2 Апаратно-програмні вимоги до проведення комп’ютерних експериментів

Для проведення комп’ютерних експериментів була використана така конфігурація апаратно-програмних ресурсів:

1. Конфігурація для Stable Diffusion Web UI від Jarvis Labs:

1.1 Версія Linux системи на момент дослідження.

1.2 Distributor ID: Ubuntu.

1.3 Description: Ubuntu 22.04.3 LTS.

1.4 Release: 22.04.

1.5 Codename: jammy.

2. Обчислювальні ресурси:

2.1 Графічний процесор: 1 x A6000 Ampere (CUDA 12.3)

2.2 Процесори: 7 CPU.

2.3 Оперативна пам'ять: 32 GB RAM.

2.4 Відеопам'ять: 48 GB VRAM.

Ця конфігурація забезпечує високу продуктивність для створення AI-згенерованих зображень, дозволяючи ефективно використовувати можливості моделі Stable Diffusion для генерації якісних результатів.

Характеристики версії AUTOMATIC1111: version: v1.9.4; python: 3.10.14; torch: 2.1.2+cu121; xformers: 0.0.23; post1; gradio: 3.41.2; checkpoint: 5493a0ec49; model: v1-5-pruned-emaonly.safetensors.

У таблицях 3.1 та 3.2 приведена інформація про орендну сесію та інформація про навчання вибірок відповідно.

Таблиця 3.1 – Інформація про орендну сесію

Дата	Device Type	No Of GPUS	No Of CPUS	RAM (GB)	VRAM (GB)	Storage (GB)	Hours Ran	Total Cost ($)
20.08.2024	A6000 Ampere	1	7	32	48	170 GB	11H:35M	$9.80

Перед початком експерименту вибірка була розширена на основі алгоритмів афінного спотворення які реалізовані за допомогою бібліотеки Rudi на мові програмування Python.

Бібліотека Rudi використовується для автоматичної аугментації (генерації варіацій) обрізаних зображень. При цьому були виконані такі операції автоматичної аугментації:

Таблиця 3.2 – Інформація про навчання на основі різних вибірок

Набір даних	Кількість зображень до розширення	Кількість зображень після розширення	Кількість кроків	Кількість епох	Загальний час навчання у (хв)	Час на один крок (хв)	Час на одну епоху (сек)
histo_fibrocystic_cystic_mastopathy	18	165	40000	242	185	0.27	45.84
Non-proliferative	31	152	40000	263	185	0.27	42.24
Proliferative	42	164	40000	244	185	0.27	45.48

1. Ймовірність трансформацій: За допомогою параметра $p = 0.7$ встановлено ймовірність 70% для кожної трансформації, тобто кожна окрема операція має шанс 70% бути застосованою до зображення.

2. Повороти: Параметри --max-left-rotation=15 та --max-right-rotation=15 задають максимальний кут обертання зображення вліво та вправо (до 15 градусів). Це додає варіативності в орієнтації зображень.

3. Масштабування: --min-factor=1 та --max-factor=1.2 визначають коефіцієнт масштабування зображення. Фактор від 1 (оригінальний розмір) до 1.2 дозволяє трохи збільшити розмір зображення, додаючи легке збільшення без великої зміни масштабу.

4. Інтенсивність спотворення: Параметр --magnitude=5 задає ступінь або інтенсивність різних застосовуваних трансформацій, таких як поворот чи масштабування, для більшої варіативності.

5. Кількість зображень: Параметр -n 125 задає кількість згенерованих варіантів для кожного зображення, щоб створити значну кількість аугментованих зразків, розширивши набір даних.

Загальний алгоритм проведення аугментації такий:

1. Скрипт спочатку обрізає зображення, щоб отримати квадратні версії, і зберігає їх.

2. Потім запускає команду rudi augment, яка використовує параметри для генерування нових зображень із випадковими змінами (поворотами, масштабуванням тощо).

3. Згенеровані аугментовані зображення переміщуються з тимчасової папки до основної папки для аугментованих зображень, де зберігаються як нові зразки разом із оригіналами.

Таким чином, Rudi автоматизує та спрощує процес створення розширеної вибірки для навчання моделей.

3.3 Результати комп’ютерних експериментів

У процесі експерименту використовувались три розширених набори даних (три класи зображень): histo_fibrocystic_cystic_mastopathy, Non-proliferative та Proliferative, кожен із яких мав від 152 до 165 зображень. Кожен набір пройшов навчання на 40,000 кроках, що забезпечувало високу деталізацію у моделюванні. Загальний час навчання для кожного з наборів склав 185 хвилин. Час на один крок залишався постійним і складав 0.27 секунди, що дозволяло досягти оптимальної швидкості обробки даних. Водночас, час на одну епоху відрізнявся

в залежності від набору, варіюючись від 42.24 до 45.84 секунди, що було зумовлено різною кількістю зображень у кожному з наборів.

На рисунку 3.5 приведені оригінальні та згенеровані зображення першого класу (histo_fibrocystic_cystic_mastopathy).

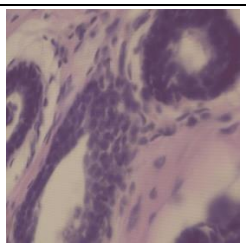
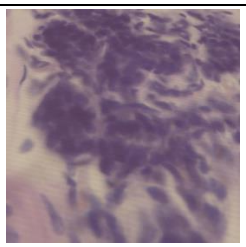
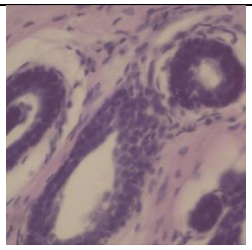
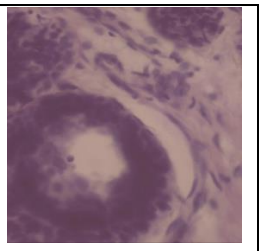
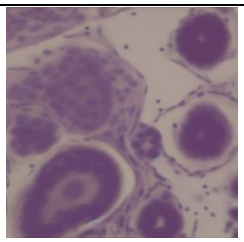
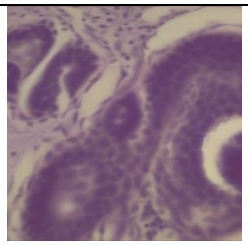
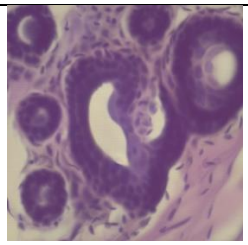
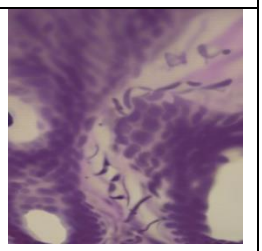
Original				
Steps	10000	20000	30000	40000
Generateed				

Рисунок 3.5 – Приклад реальних і згенерованих зображень першого класу

На рисунку 3.6 приведені результати тестування згенерованих зображень на основі метрик FID та IS.

Приведемо результати експерименту згенерованих зображень;

– FID (Frechet Inception Distance): Зі збільшенням ітерацій значення FID змінюється, але загалом тримається в межах 0.775–0.925. Найменше значення спостерігається на 5 тисячах ітерацій, а найбільше на 20 тисячах.

– IS (Inception Score): Показник, що вимірює різноманітність і реалістичність зображень, згенерованих моделлю. Значення IS варіюється в межах 2.1–2.4. Максимальне значення спостерігається на 15 тисячах ітерацій, а мінімальне – на 30 тисячах.

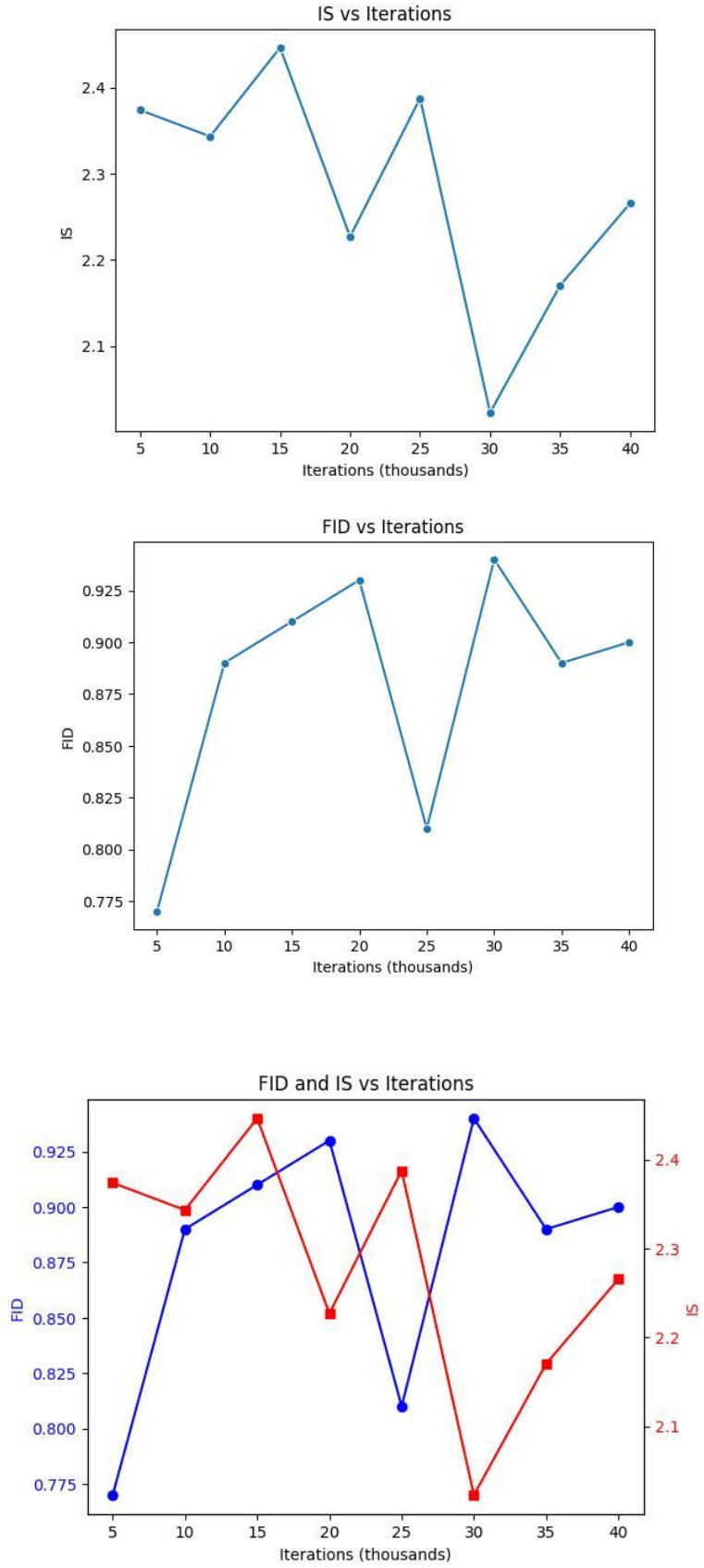


Рисунок 3.6 – Результати тестування згенерованих зображень першого класу на основі метрик FID,IS

У таблиці 3.3 приведені значення метрик FID,IS від кількості ітерацій навчання.

Таблиця 3.3 – Результати експериментів

Iteration (thousands)	FID	IS
5	0.775	2.35
10	0.875	2.3
15	0.9	2.4
20	0.925	2.25
25	0.875	2.35
30	0.8	2.1
35	0.9	2.15
40	0.875	2.25

На рисунку 3.7 приведені приклади реальних і згенерованих зображень для другого класу (Proliferative).

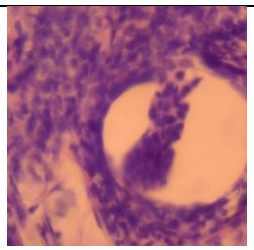
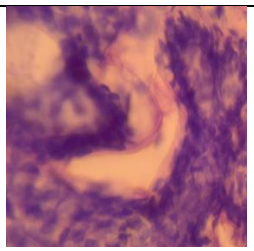
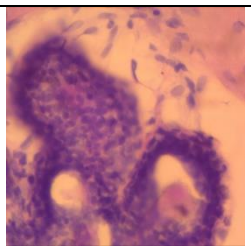
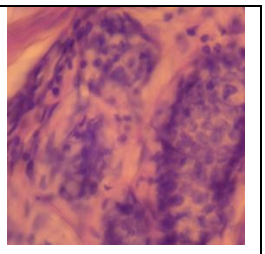
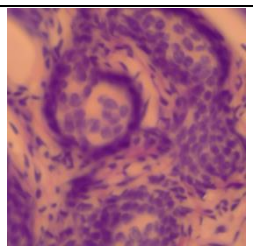
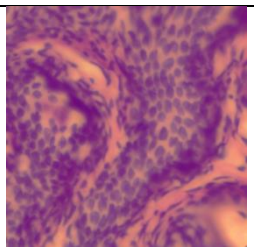
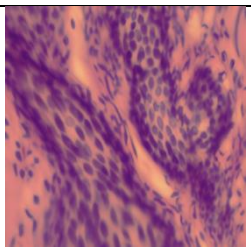
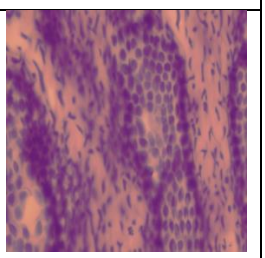
Original				
Steps	10000	20000	30000	40000
Generateed				

Рисунок 3.7 – Приклад реальних і згенерованих зображень другого класу (Proliferative)

На рисунку 3.8 приведені результати тестування зображень другого класу на основі метрик IS, FID.

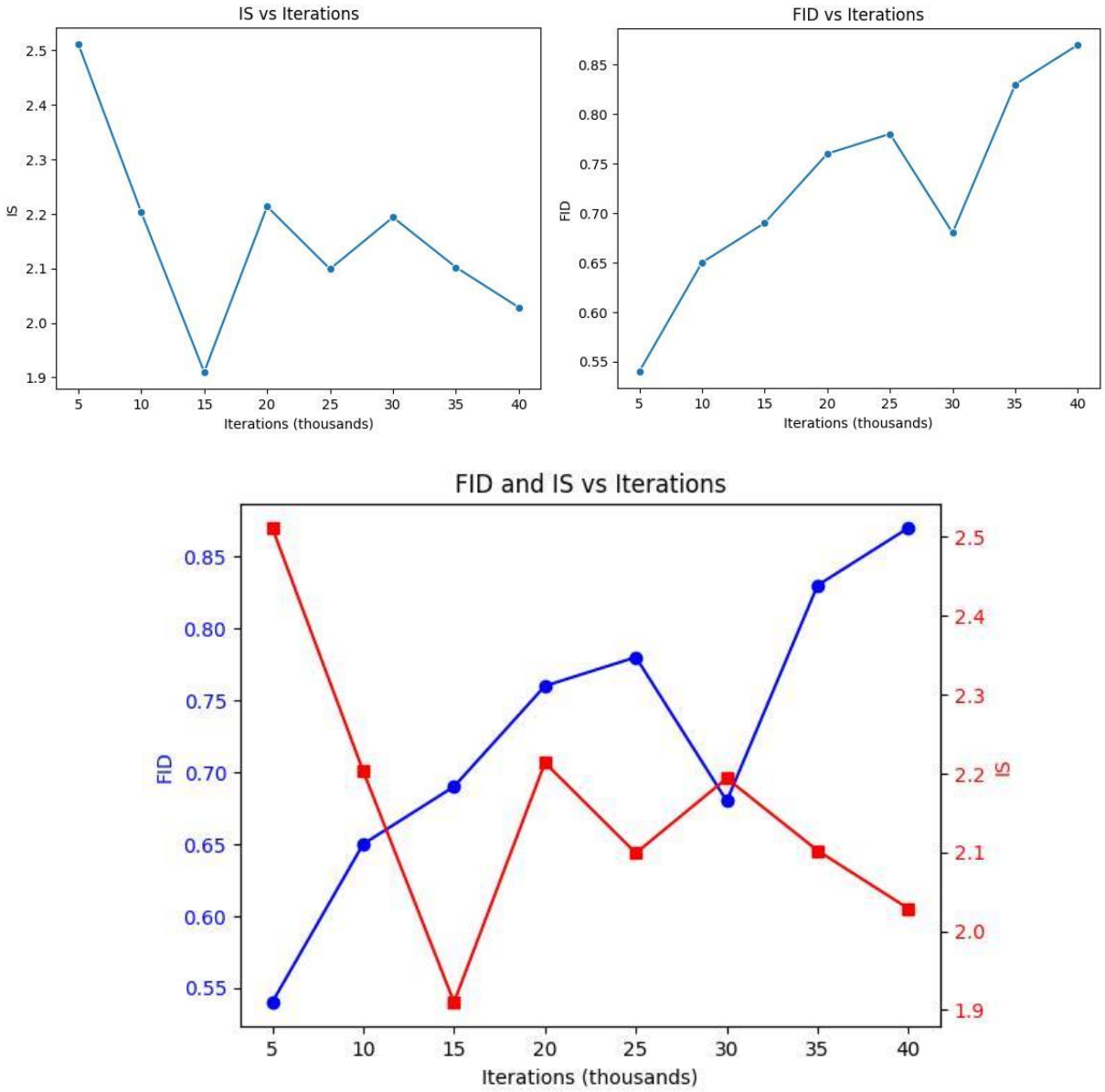


Рисунок 3.8 – Результати тестування згенерованих зображень другого класу на основі метрик FID, IS

Результати експерименту із зображеннями другого класу у метриках FID,IS такі:

– FID (Frechet Inception Distance): У нових даних видно загальну тенденцію до зростання FID від 0.55 на 5 тисячах ітерацій до 0.85 на 40 тисячах.

– IS (Inception Score): IS показує спадний тренд. Значення починається з 2.5 на 5 тисячах ітерацій і поступово зменшується до 1.95 на 40 тисячах ітерацій.

У таблиці 3.4 приведені значення метрик FID,IS від кількості ітерацій навчання для зображень другого класу.

Таблиця 3.4 – Результати експериментів

Iteration (thousands)	FID	IS
5	0.55	2.5
10	0.65	2.2
15	0.7	1.95
20	0.75	2.15
25	0.725	2.05
30	0.675	2.1
35	0.8	2.05
40	0.85	1.95

На рисунку 3.9 приведені приклади реальних і згенерованих зображень для третього класу (non-proliferative)

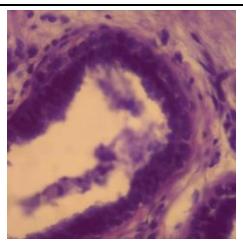
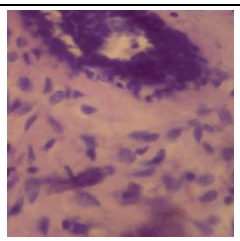
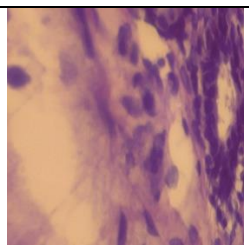
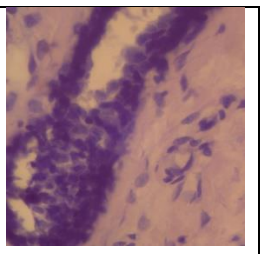
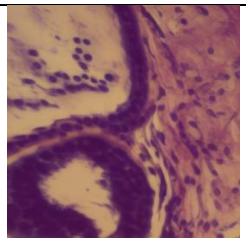
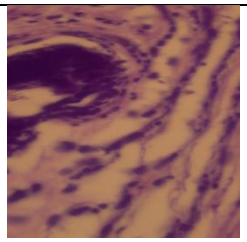
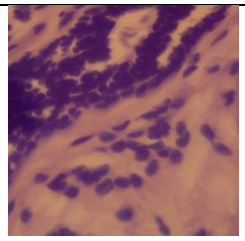
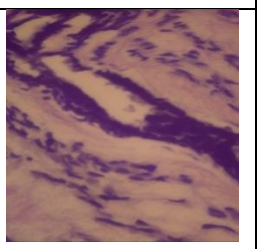
Original				
Steps	10000	20000	30000	40000
Generateed				

Рисунок 3.9 –Приклади реальних і згенерованих зображень для третього класу

Результати оцінки якості згенерованих зображень на основі метрик FID,IS приведені на рисунку 3.10.

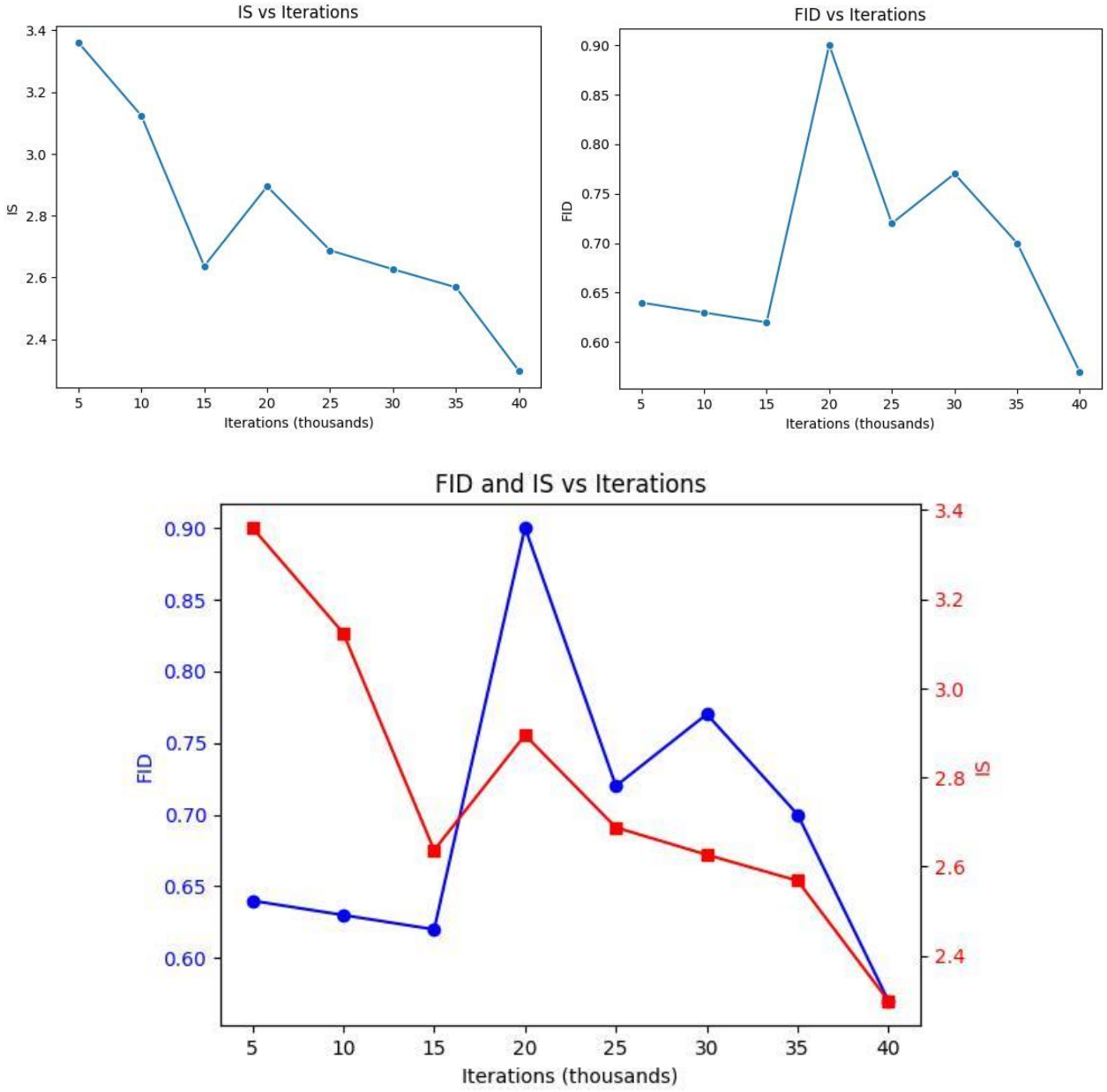


Рисунок 3.10 – Результати тестування згенерованих зображень третього класу на основі метрик FID, IS

Результати експерименту із зображеннями третього класу на основі метрик FID,IS такі:

– FID (Frechet Inception Distance): На початку ітерацій значення FID починається з 0.6 і зростає до пікового значення 0.9 на 20 тисячах ітерацій, після

чого знову поступово знижується до 0.6 на 40 тисячах. Це може свідчити про нестабільність в якості генерації на певних етапах тренування, але кінцеве значення на 40 тисячах показує покращення, порівняно з середніми значеннями.

– IS (Inception Score): Значення IS демонструє спадний тренд, починаючи з 3.4 на 5 тисячах ітерацій і знижуючись до 2.3 на 40 тисячах. Це може вказувати на поступове зниження різноманітності чи якості зображень з кожним етапом тренування.

У таблиці 3.5 приведені значення метрик FID,IS від кількості ітерацій навчання для зображень третього класу.

Таблиця 3.5 – Результати експериментів

Iteration (thousands)	FID	IS
5	0.6	3.4
10	0.625	3.2
15	0.65	2.9
20	0.9	2.7
25	0.7	2.8
30	0.75	2.6
35	0.7	2.5
40	0.6	2.3

На рисунку 3.11 приведені загальні графіки для трьох класів для метрик FID, IS.

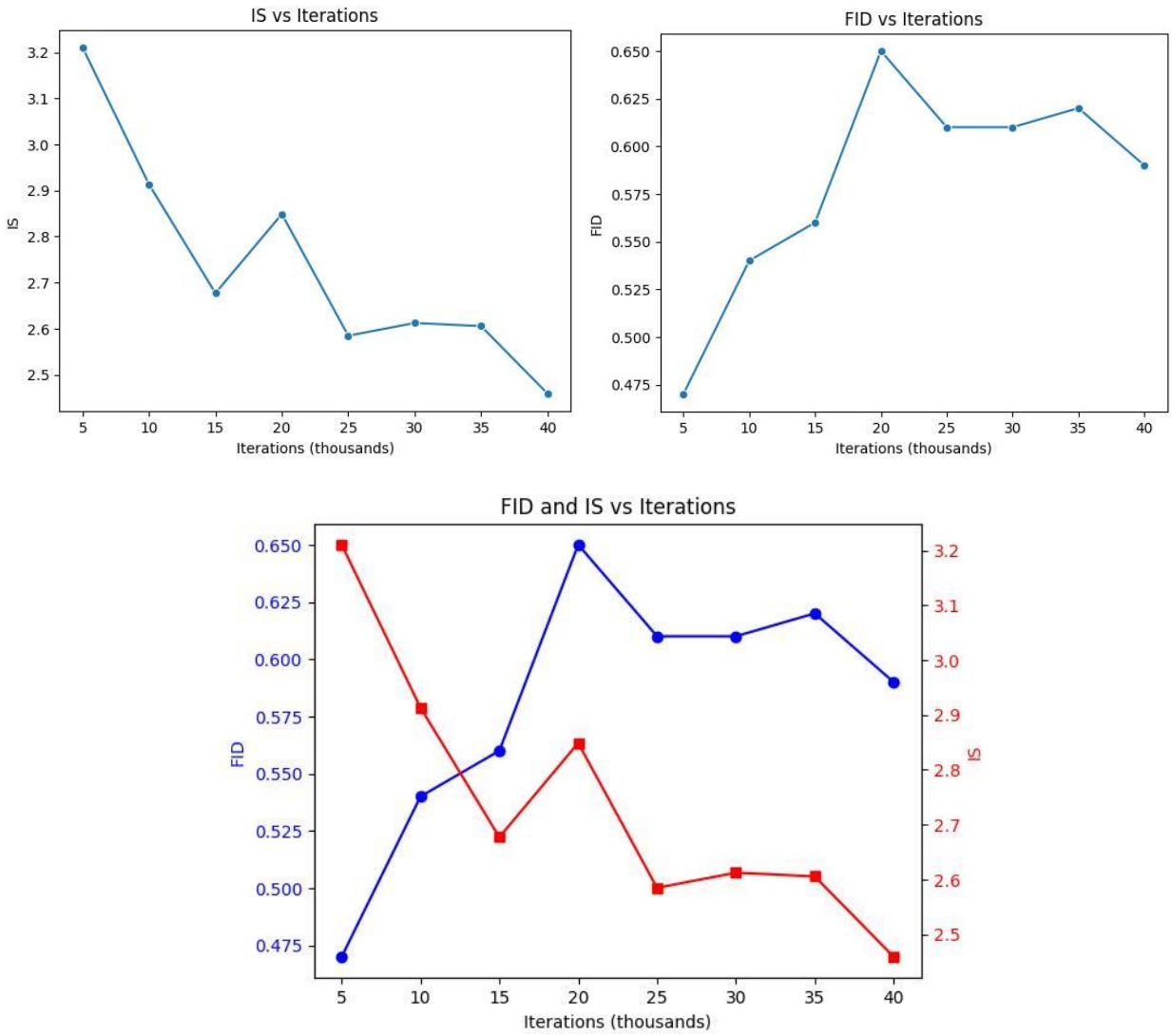


Рисунок 3.11 – Результати тестування згенерованих зображень по трьох класах на основі метрик FID,IS

У таблиці 3.6 приведені значення метрик FID,IS від кількості ітерацій навчання для трьох класів.

Таблиця 3.6 – Результати експериментів

Iteration (thousands)	FID	IS
5	0.475	3.2
10	0.525	2.9
15	0.55	2.8
20	0.65	2.7

Продовження таблиці 3.6

25	0.6	2.6
30	0.6	2.55
35	0.625	2.5
40	0.55	2.45

Результати експерименту із зображеннями трьох класів на основі метрик FID,IS такі:

- FID (Frechet Inception Distance): Показник FID поступово зростає з 5 тисяч ітерацій, досягаючи піку на 20 тисячах (0.65), після чого починає коливатись, а потім знижується до 0.55 на 40 тисячах. Це може означати, що після 20 тисяч ітерацій модель намагається стабілізувати якість зображень, проте покращення спостерігається ближче до кінця.
- IS (Inception Score): Значення IS поступово знижується протягом усіх ітерацій, починаючи з 3.2 на 5 тисячах ітерацій і знижуючись до 2.45 на 40 тисячах.

3.4 Висновки до розділу 3

У третьому розділі описано інтерфейс програмної системи Stable Diffusion у трьох режимах тренування, генерування і навчання мережі Hypernetwork. Приведені вимоги до апаратно-програмних засобів генерування гістологічних зображень. На основі цих програмно-апаратних засобів згенеровані гістологічні зображення для трьох класів і зроблено оцінку їх якості на основі метрик FID,IS.

ВИСНОВКИ

1. Проведено аналіз сучасних тенденцій у штучному інтелекті і показано, що генеративний інтелект є актуальним напрямком, методи якого використовуються для генерування зображень, звуку, тексту.

2. Проаналізовано підходи генеративного інтелекту для синтезу зображень і показано, що дифузійні моделі є ефективним для генерування біомедичних зображень.

3. Здійснено аналіз програмних засобів і обґрунтовано використання програмного засобу Stable Diffusion для генерування біомедичних зображень.

4. Досліджено дифузійні алгоритми синтезу зображень і розроблено узагальнений алгоритм синтезу гістологічних зображень, що дало можливість провести обчислювальні експерименти в середовищі Stable Diffusion.

5. Описано інтерфейс програмного середовища Stable Diffusion, завдяки чому було проведено обчислювальні експерименти для генерування гістологічних зображень.

6. Проведено обчислювальні експерименти в середовищі Stable Diffusion з використанням трьох класів реальних гістологічних зображень і отримано розширений штучний набір даних гістологічних зображень.

7. Якість синтезованих зображень за метрикою FID починається з 0.6 і зростає до значення 0.9 на 20 тисячах ітерацій, після чого знову поступово знижується до 0.6 на 40 тисячах. За метрикою IS значення починаються з 3.4 на 5 тисячах ітерацій і знижуючись до 2.3 на 40 тисячах. Це вказує на поступове зниження різноманітності та якості зображень з кожним етапом тренування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Домбровський М. О. Синтез зображень на основі генеративного інтелекту. Інтелектуальні комп’ютерні системи та мережі : тези доп. VIII Наук.-практ. конф. молодих вчених і студентів (5 груд. 2023 р.). Тернопіль : ЗУНУ, 2023. С. 28.

2. Домбровський М. О. Синтез біомедичних зображень на основі дифузійно – імовірнісних моделей. Інтелектуальні комп’ютерні системи та мережі : тези доп. IX Наук.-практ. конф. студентів, аспірантів, молодих вчених (21 трав. 2024 р.). Тернопіль : ЗУНУ, 2024. С. 34.

3. Berezsky O., Liashchynskyi P., Melnyk G., Dombrovskyi M. and Berezkyi M. Synthesis of biomedical images based on generative intelligence tools. 7th International Conference on Informatics & Data-Driven Medicine. University of Birmingham, Birmingham, United Kingdom. November 14 - 16, 2024. (Scopus, Web of Science).

4. Березький О. М., Дубчак Л. О., Мельник Г. М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Магістр”. Спеціальність: 123 – Комп’ютерна інженерія. Магістерська програма – «Комп’ютерна інженерія». Тернопіль: ЗУНУ, 2021. 32 с.

5. Методи, алгоритми та програмні засоби опрацювання біомедичних зображень / Березький О. М., Батько Ю.М., Березька К.М., Вербовий С.О., Дацко Т.В., Дубчак Л.О., Ігнатєв І.В., Мельник Г.М., Николюк В.Д., Піцун О.Й. – Тернопіль: Економічна думка, ТНЕУ, 2017. – 330 с.

6. Березький О.М., Піцун О.Й., Боднар А. Р., Долинюк Т.М. Класифікація гістологічних та цитологічних зображень на основі згорткових нейронних мереж. Штучний інтелект. Київ, 2017. №1 (75). С. 33-42.

7. Березький О. М., Мельник Г.М. Інформаційна технологія аналізу та синтезу гістологічних зображень. Матер. Одинадцятої всеукр. міжнар. конф.

«Оброблення сигналів і зображень та розпізнавання образів» (УкрОБРАЗ’2012), Київ, 15-19 жовтня 2012 р. К., 2012. С. 161-164.

8. Березький О. М., Мельник Г.М., Дацко Т.В., Вербовий С. О. База даних цитологічних та гістологічних зображень ауто- та ксеногенних тканин. Науковий вісник НЛТУ України: зб. наук.-техн. праць. Львів: РВВ НЛТУ України. 2014. Вип. 24.10. С. 338-345.

9. Березький О., Піцун О., Дацко Т., Дериш Б., Мельник Г. База даних імуногістохімічних зображень раку молочної залози. Комп’ютерні системи та інформаційні технології. №1, 2022. С. 75-82. DOI: https://doi.org/10.31891/CSIT-2022-1-10

10. Березский О. Н., Березская Е. Н. Количественная оценка качества сегментации изображений на основе метрик. Управляющие системы и машины. 2015. №6. С.59-65.

11. Березький О. М., Батько Ю.М., Мельник Г.М. Методи сегментації біомедичних зображень. Вісник Хмельницького національного університету. Технічні науки. 2010. №1. С.189- 197.

12. Сеньо П. С., Випадкові процеси : підручник / П. С. Сеньо ; Мін-во освіти і науки України, Львів. нац. ун-т ім. І. Франка.- Л., 2006. 288 с.

13. Kazerouni A., Aghdam E.K., Heidari M., Azad R., Fayyaz M., Hacihaliloglu I., Merhof D. Diffusion models in medical imaging: A comprehensive survey // Medical Image Analysis. – 2023. – Vol. 88. – P. 102846. DOI: https://doi.org/10.1016/j.media.2023.102846.

14. Pozzi M., Noei S., Robbi E., Cima L., Moroni M., Munari E., Torresani E., Jurman G. Generating synthetic data in digital pathology through diffusion models: a multifaceted approach to evaluation // medRxiv. – 2023. DOI: https://doi.org/10.1101/2023.11.21.23298808.

15. Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B. High-Resolution Image Synthesis with Latent Diffusion Models // arXiv. – 2021. – arXiv:2112.10752. DOI: https://doi.org/10.48550/arXiv.2112.10752.

16. Moghadam P.A., Van Dalen S., Martin K.C., Lennerz J., Yip S., Farahani H., Bashashati A. A Morphology Focused Diffusion Probabilistic Model for Synthesis of Histopathology Images // arXiv. – 2022. – arXiv:2209.13167. DOI: https://doi.org/10.48550/arXiv.2209.13167.

17. Ho J., Saharia C., Chan W., Fleet D.F., Norouzi M., Salimans T. Cascaded Diffusion Models for High Fidelity Image Generation // arXiv. – 2021. – arXiv:2106.15282. DOI: https://doi.org/10.48550/arXiv.2106.15282.

18. Yang L., Zhang Z., Song Y., Hong S., Xu R., Zhao Y., Zhang W., Cui B., Yang M.-H. Diffusion Models: A Comprehensive Survey of Methods and Applications // arXiv. – 2022. – arXiv:2209.00796. DOI: https://doi.org/10.48550/arXiv.2209.00796.

19. Chen M., Mei S., Fan J., Wang M. An Overview of Diffusion Models: Applications, Guided Generation, Statistical Rates and Optimization // arXiv. – 2024. – arXiv:2404.07771. DOI: https://doi.org/10.48550/arXiv.2404.07771.

20. Березький О. М., Березька К. М., Попіна С. Ю. Статистичне оброблення цитологічних зображень. Вісник Хмельницького національного університету: зб. наук.-техн. праць. Сер.: Технічні науки. 2012. № 5. С. 161–164.

21. Zhou R., Jiang C., Xu Q. A survey on generative adversarial network-based text-to-image synthesis // Neurocomputing. – 2021. – Vol. 451. – P. 316–336. DOI: https://doi.org/10.1016/j.neucom.2021.04.069.

22. Xu Y., Liang J., Zhuo Y., Liu L., Xiao Y., Zhou L. TDASD: Generating medically significant fine-grained lung adenocarcinoma nodule CT images based on stable diffusion models with limited sample size // Computer Methods and Programs in Biomedicine. – 2024. – Vol. 248. – P. 108103. DOI: https://doi.org/10.1016/j.cmpb.2024.108103.

23. Zhang H., Xu H., Tian X., Jiang J., Ma J. Image fusion meets deep learning: A survey and perspective // Information Fusion. – 2021. – Vol. 76. – P. 323–336. DOI: https://doi.org/10.1016/j.inffus.2021.06.008.

24. Sohl-Dickstein J., Weiss E.A., Maheswaranathan N., Ganguli S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics // arXiv. – 2015. – arXiv:1503.03585. DOI: https://doi.org/10.48550/arXiv.1503.03585.

25. Ho J., Jain A., Abbeel P. Denoising Diffusion Probabilistic Models // arXiv. – 2020. – arXiv:2006.11239. DOI: https://doi.org/10.48550/arXiv.2006.11239.

26. Song J., Meng C., Ermon S. Denoising Diffusion Implicit Models // arXiv. – 2020. – arXiv:2010.02502. DOI: https://doi.org/10.48550/arXiv.2010.02502.

27. Nichol A., Dhariwal P. Improved Denoising Diffusion Probabilistic Models // arXiv. – 2021. – arXiv:2102.09672. DOI: https://doi.org/10.48550/arXiv.2102.09672.

28. Oko K., Akiyama S., Suzuki T. Diffusion Models are Minimax Optimal Distribution Estimators // arXiv. – 2023. – arXiv:2303.01861. DOI: https://doi.org/10.48550/arXiv.2303.01861.

29. Denich E. Error estimates for a Gaussian rule involving Bessel functions // arXiv. – 2023. – arXiv:2301.11738. DOI: https://doi.org/10.48550/arXiv.2301.11738.

30. Єрьоменко В.О., Шинкарик М.І., Бабій Р. М., Процик А. І. Практикум з теорії імовірностей та математичної статистики: навч. посіб. Тернопіль : Економічна думка, 2005. 317 с.

31. Song Y., Sohl-Dickstein J., Kingma D.P., Kumar A., Ermon S., Poole B. Score-Based Generative Modeling through Stochastic Differential Equations // arXiv. – 2020. – arXiv:2011.13456. DOI: https://doi.org/10.48550/arXiv.2011.13456.

32. Berezsky O., Pitsun O., Derysh B., Datsko T., Berezka K., Savka N. Automatic segmentation of immunohistochemical images based on U-NET architectures. IDDM-2021: 4th International Conference on Informatics & Data-Driven Medicine, November 19–21, 2021 Valencia, Spain. pp. 22-33, http://ceur-ws.org/Vol-3038/paper3.pdf

33. Berezsky, O.; Pitsun, O.; Berezkyy, M.; Batko, Y.; Melnyk, G. MLOps Approach for Automatic Image Segmentation Based on U-Net. Preprints 2024, 2024072447. https://doi.org/10.20944/preprints202407.2447.v1.

34. Неміш В. М., Процик А. І., Березька К. М. Практикум з вищої математики: Навчальний посібник. Тернопіль: Економічна думка, 2007. 302 с.

35. Березький О. М., Батько Ю.М., Мельник Г.М. Комп'ютерна система аналізу біомедичних зображень. Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. 2009. № 570. С. 84-89.

36. Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis // arXiv. – 2021. – arXiv:2105.05233. DOI: https://doi.org/10.48550/arXiv.2105.05233.

37. Luo C. Understanding Diffusion Models: A Unified Perspective // arXiv. – 2022. – arXiv:2208.11970. DOI: https://doi.org/10.48550/arXiv.2208.11970.

38. Li P., Zhong L., Zhang H., Bian J. On the Generalization Properties of Diffusion Models // arXiv. – 2023. – arXiv:2311.01797. DOI: https://doi.org/10.48550/arXiv.2311.01797.

39. Marjieh R., Sucholutsky I., Langlois T.A., Jacoby N., Griffiths T.L. Analyzing Diffusion as Serial Reproduction // arXiv. – 2022. – arXiv:2209.14821. DOI: https://doi.org/10.48550/arXiv.2209.14821.

40. Peebles W., Xie S. Scalable Diffusion Models with Transformers // arXiv. – 2023. – arXiv:2212.09748. DOI: https://doi.org/10.48550/arXiv.2212.09748.

41. Liu Z., Luo D., Xu Y., Jaakkola T., Tegmark M. GenPhys: From Physical Processes to Generative Models // arXiv. – 2023. – arXiv:2304.02637. DOI: https://doi.org/10.48550/arXiv.2304.02637.

42. Gulrajani I., Hashimoto T.B. Likelihood-Based Diffusion Language Models // arXiv. – 2023. – arXiv:2305.18619. DOI: https://doi.org/10.48550/arXiv.2305.18619.

43. Berezsky O., Zarichnyi M., Pitsun O. Development of a metric and the methods for quantitative estimation of the segmentation of biomedical images. Eastern-European Journal of Enterprise Technologies. 2017. V. 6, № 4. P. 4–11. https://doi.org/10.15587/1729-4061.2017.119493