Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

магістра
(назва освітньо-кваліфікаційного рівня)

студента	*Ніколаєнка Артема Віталійовича*
	(ПІБ)
академічної групи	*122М-22-3*
	(шифр)
спеціальності	*122 Комп'ютерні науки*
	(код і назва спеціальності)
освітньої програми	*«122 Комп'ютерні науки»*
	(назва освітньої програми)
на тему:	*Дослідження та застосування методу Stable Diffusion*
на базі штучного інтелекту в контексті генеративного мистецтва	

_____*А.В. Ніколаєнко*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
розділів кваліфікаційної роботи				
спеціальний	*доц. Спірінцев В.В.*			
економічний				
Рецензент				
Нормоконтролер	*проф. Лактіонов І.С.*			

Дніпро
2023

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
Завідувач кафедри
Програмного забезпечення комп'ютерних систем
(повна назва)

_____ М.О. Алексєєв
(підпис) (прізвище, ініціали)

« » _____ 20 23 року

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

студенту *122м-22-3* *Ніколаєнко Артему Віталійовичу*
(група) (прізвище та ініціали)

Тема кваліфікаційної роботи *Дослідження та застосування методу Stable Diffusion на базі штучного інтелекту в контексті генеративного Мистецтва*

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 09.10.2023 р. № 1227-с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес генерації зображень із використанням моделей латентної дифузії.

Предмет досліджень – інформаційні та структурні моделі процесу генерації зображень з використанням моделей латентної дифузії. Дослідження теоретико-прикладних засад створення та функціонування даних моделей та основи налаштування параметрів їх використання в практичних завданнях генеративного мистецтва.

Мета НДР – підвищення якості вихідних зображень та ефективності підбору параметрів генерації при зменшенні необхідних обчислювальних ресурсів, що задіяні у процесі генерації зображень із використанням моделей латентної дифузії.

Вихідні дані для проведення роботи – теоретичні та експериментальні дослідження, основи процесу генерації зображень з використанням моделей латентної дифузії у контексті генеративного мистецтва.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Новизна запропонованих рішень. Отримав подальший розвиток напрямок генеративного мистецтва, що базується на застосуванні моделей латентної дифузії при генерації зображень. Запропонований підхід дозволяє покращити якість генерованих зображень та пришвидшити робочий процес, прискоривши етап підбору оптимальних параметрів генерації при зменшенні необхідної кількості обчислювальної потужності.

Практична цінність результатів полягає у тому, що сформовані у роботі рекомендації та вказівки дозволяють користувачам покращити якість генерованих зображень та оптимізувати процес підбору параметрів генерації із зменшенням кількості необхідних обчислювальних ресурсів при генерації зображень з використанням моделей латентної дифузії у завданнях генеративного мистецтва.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє побачити, оцінити та використати отримані рекомендації у процесі генерації зображень із використанням моделей латентної дифузії. В результаті роботи повинен бути сформований комплекс конкретних рекомендації для оптимізації процесу налаштування параметрів генерації при роботі із моделями латентної дифузії з метою покращення якості генерованих зображень та ефективності використання обчислювальних ресурсів, задіяних під час генерації. Отримані рекомендації є універсальними для будь-яких систем генерації зображень, які використовують моделі латентної дифузії.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	**Строки виконання робіт (початок – Кінець**
Аналіз теми та постановка задачі	09.10.2023-25.10.2023
Аналіз підходів до вирішення задач генеративного мистецтва: генеративні моделі та їх архітектура, особливості їх використання. Теоретичне дослідження моделей латентної дифузії, їх алгоритмів навчання та використання у завданнях генерації зображень	26.10.2023-13.11.2023
Експерименти та практичні дослідження із використанням моделей латентної дифузії, аналіз отриманих результатів та створення комплексу рекомендацій для оптимізації етапу підбору параметрів генерації	14.11.2023-03.12.2023

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від реалізації результатів роботи очікується позитивним завдяки скороченню часу, який потрібно витратити на отримання зображень бажаної якості, і відповідно, скороченню витрат на задіяні обчислювальні ресурси.

Соціальний ефект від реалізації результатів роботи очікується позитивним завдяки оптимізації та пришвидшенню процесу генерації зображень, що дозволить скоротити робочий час, необхідний на створення зображень бажаної якості.

7 ДОДАТКОВІ ВИМОГИ

Завдання видав _____ *Спірінцев В.В.*
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ *Ніколаєнко А.В.*
(підпис) (прізвище, ініціали)

Дата видачі завдання: *09.10.2023 р.*

Термін подання кваліфікаційної роботи до ЕК *04.12.2023*

РЕФЕРАТ

Пояснювальна записка: 147 стор., 54 рис., 3 табл., 3 додатки, 47 джерел.

Об'єкт дослідження: процес генерації зображень із використанням моделей латентної дифузії.

Предмет дослідження: інформаційні та структурні моделі процесу генерації зображень з використанням моделей латентної дифузії. Дослідження теоретико-прикладних засад створення та функціонування даних моделей та основи налаштування параметрів їх використання в практичних завданнях генеративного мистецтва.

Мета роботи: підвищення якості вихідних зображень та швидкості налаштування робочого процесу генерації зображень при зменшенні кількості необхідних обчислювальних ресурсів із використанням моделей латентної дифузії.

Методи дослідження. Методи дослідження базуються на основних принципах роботи моделей латентної дифузії. Використано методи теоретичного моделювання, теоретичні основи процесу генерації зображень із використанням моделей латентної дифузії, метод оцінки отриманих зображень людськими експертами та практичне використання моделей латентної дифузії.

Новизна запропонованих рішень. Отримав подальший розвиток напрямок генеративного мистецтва, що базується на застосуванні моделей латентної дифузії при генерації зображень. Запропонований підхід дозволяє покращити якість генерованих зображень та пришвидшити робочий процес, прискоривши етап підбору оптимальних параметрів генерації при зменшенні необхідної кількості обчислювальної потужності.

Практична цінність результатів полягає в тому, що сформовані у роботі методи дозволяють користувачам оптимізувати робочий процес та покращити якість генерованих зображень при використанні моделей латентної дифузії.

Область застосування. Розроблені рекомендації та вказівки можуть бути практично використані у задачах генеративного мистецтва.

Значення роботи та висновки. Розроблені рекомендації та вказівки дозволяють підвищити якість вихідних зображень, оптимізувати та пришвидшити робочий процес та зменшити кількісті необхідних обчислювальних ресурсів при роботі із моделями латентної дифузії у задачах генеративного мистецтва.

Прогнози щодо розвитку досліджень. Впровадження сформованих рекомендацій та вказівок надасть змогу скоротити витрати на необхідні обчислювальні ресурси та збільшити якість генерованих зображень за рахунок оптимізації процесу підбору оптимальних параметрів генерації при роботі із моделями латентної дифузії у задачах генеративного мистецтва.

Список ключових слів: моделі латентної дифузії, latent diffusion models, машинне навчання, генеративні моделі, генеративне мистецтво, генерація зображень, оптимізація параметрів генерації, Stable Diffusion, ControlNet.

ABSTRACT

Explanatory note: 147 pages, 54 figures, 3 tables, 3 appendices, 47 sources.

Object of research: the process of image generation using latent diffusion models.

Subject of research: information and structural models of the image generation process using latent diffusion models. Research of theoretical and applied principles of creation and functioning of these models and the basis for setting the parameters of their use in practical tasks of generative art.

Purpose of Master's thesis: to improve the quality of output images and the speed of setting up the image generation workflow while reducing the amount of computing resources required using latent diffusion models.

Research methods. The research methods are based on the basic principles of latent diffusion models. The methods used are theoretical modeling, theoretical foundations of the image generation process using latent diffusion models, a method for evaluating the obtained images by human experts, and the practical use of latent diffusion models.

Originality of research has been further developed in the direction of generative art based on the use of latent diffusion models in image generation. The proposed approach improves the quality of generated images and speeds up the workflow by accelerating the stage of selecting optimal generation parameters while reducing the amount of computing power required.

Practical value of the results is that the methods developed in this work allow users to optimize the workflow and improve the quality of generated images when using latent diffusion models.

Scope of application. The developed recommendations and guidelines can be practically used in the tasks of generative art.

Significance of the work and conclusions. The developed recommendations and guidelines allow us to improve the quality of source images, optimize and speed up the workflow, and reduce the amount of computing resources required when working with latent diffusion models in generative art tasks.

Research forecast and development. Implementation of the developed recommendations and guidelines will reduce the cost of the necessary computing resources and improve the quality of the generated images by optimizing the process of selecting the optimal generation parameters when working with latent diffusion models in generative art problems.

Keywords: latent diffusion models, machine learning, generative models, generative art, image generation, optimization of generation parameters, Stable Diffusion, ControlNet

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

МН – машинне навчання

МД – моделі дифузії

МЛД – моделі латентної дифузії

ПЗ – програмне забезпечення

LDM – latent diffusion model

SD – Stable Diffusion

UI – User Interface

GPU - Graphics processing unit

CU – computing units

ЗМІСТ

ВСТУП

Актуальність теми. Генерація зображень з використанням алгоритмів та моделей машинного навчання – це одна з ключових тем, яка зараз знаходиться у центрі уваги науковців та творців. На даний момент існує множина методів та підходів, які можуть бути використані у задачах генеративного мистецтва, однак, усі вони мають різну якість вихідних результатів та ефективність відносно до задіяних обчислювальних ресурсів.

Тож вибір інструментів та методологій для вирішення завдань генеративного мистецтва є дуже важливим етапом, на який впливають кілька факторів, такі якість та швидкість генерації зображень, вимоги до технічних характеристик обчислювальної машини та ефективність їх використання. Вибір генеративної моделі залежить від бюджету, технічних характеристик ЕОМ та вимог до генерованих зображень.

Технічні вимоги до обчислювальної машини, на якій виконується генерація зображень, є ключовим обмежуючим фактором. Абсолютна більшість наявних генеративних моделей потребує для своєї роботи значної обчислювальної потужності, такої як, наприклад, обсяг відеопам’яті, швидкість ядра та кількість одиниць обчислення графічного процесора, які зазвичай наявні тільки на потужних комерційних серверах, що робить їх використанням коштовним, а процес налаштування параметрів генерації менш гнучким та доступним для користувача.

Через наведені фактори, до появи моделей латентної дифузії, масове самостійне дослідження та використання генеративних моделей не було доступним та зручним для звичайного користувача. Моделі латентної дифузії стали інноваційним рішенням у завданнях генеративного мистецтва через ряд таких переваг, як висока якість вихідних результатів, гнучкість налаштування та їх нова архітектура, яка призвела до суттєвого зниження вимог до технічних характеристик ЕОМ та збільшення ефективності їх використання у обчислювальних процесах, задіяних під час генерації.

Мета і задачі дослідження. Метою роботи є дослідження ефективності моделей латентної дифузії, як інструменту для створення якісних зображень у завданнях генеративного мистецтва.

Для досягнення мети дослідження необхідно розв’язати наступні задачі:

1. Провести аналіз існуючих технологій, які дозволяють виконувати генерацію зображень (GAN, VAE, дифузійні моделі).

2. Здійснити порівняння моделей латентної дифузії із іншими генеративними моделями та дослідити переваги використання даного класу моделей, відносно інших існуючих варіантів.

3. Теоретично дослідити та проаналізувати принципи роботи моделей латентної дифузії, їх архітектуру, алгоритми навчання та використання, параметри генерації та можливості застосування даного класу генеративних моделей у задачах генеративного мистецтва.

4. Провести експериментальні дослідження процесу генерації зображень із використанням моделей латентної дифузії та проаналізувати отримані під час практичного дослідження результати.

5. Сформувати перелік рекомендації для оптимального налаштування параметрів під час генерації зображень із використанням моделей латентної дифузії, які дозволять покращити якість вихідних зображень, пришвидшити та оптимізувати робочий процес та зменшити кількість необхідних обчислювальних ресурсів.

Об'єктом дослідження є процес генерації зображень із використанням моделей латентної дифузії.

Предметом дослідження є інформаційні та структурні моделі процесу роботи моделей латентної дифузії. Дослідження теоретико-прикладних засад створення та функціонування даних моделей та основи налаштування їх параметрів генерації для використання в практичних завданнях генеративного мистецтва.

Методи дослідження. Методи дослідження базуються на основних принципах роботи моделей латентної дифузії. Використано методи теоретичного

моделювання, теоретичні основи процесу генерації зображень із використанням моделей латентної дифузії, метод оцінки отриманих зображень людськими експертами та практичне використання моделей латентної дифузії

Наукова новизна роботи. Отримав подальший розвиток напрямок генеративного мистецтва, що базується на застосуванні моделей латентної дифузії при генерації зображень. Запропонований підхід дозволяє покращити якість генерованих зображень та пришвидшити робочий процес, прискоривши етап підбору оптимальних параметрів генерації при зменшенні необхідної кількості обчислювальної потужності.

Практичне значення отриманих в роботі результатів полягає у тому, що сформовані у роботі рекомендації та вказівки дозволяють користувачам покращити якість генерованих зображень та оптимізувати процес підбору параметрів генерації із зменшенням кількості необхідних обчислювальних ресурсів при генерації зображень зі використанням моделей латентної дифузії у завданнях генеративного мистецтва

Особистий внесок автора. Було теоретично та практично досліджено процес використання моделей латентної дифузії, систематизовано знання про алгоритми їх навчання та використання, архітектуру, параметри генерації та можливості застосування даних моделей у контексті генеративного мистецтва. Із метою підвищення якості, швидкості генерації зображень та зменшення необхідних обчислювальних ресурсів було сформовано перелік рекомендацій та вказівок для налаштування параметрів генерації зображень при роботі із моделями латентної дифузії.

Структура та обсяг кваліфікаційної роботи. Відповідно до мети, завдань і предмета дослідження, кваліфікаційна робота складається з реферату, вступу, трьох основних розділів і висновків, списку використаних джерел та 4 додатків. Загальний обсяг роботи містить 147 сторінок друкованого тексту, із них основної частини – 117 сторінок з 54 рис., спеціальної – 95 сторінок, списку використаних джерел з 47 найменуванням на 4 сторінках, 4 додатки на 13 сторінках.

РОЗДІЛ 1
АНАЛІЗ ПОТОЧНИХ ЗАСОБІВ ТА ІНСТРУМЕНТІВ У СФЕРІ ГЕНЕРАТИВНОГО МИСТЕЦТВА

1.1. Основи генерації зображень

1.1.1. Введення до генерації зображень

Генерація зображень є важливою областю у сучасному світі технологій та сфері машинного навчання, яка знаходить застосування в різних галузях, від комп'ютерної графіки та мистецтва до медицини та наукових досліджень. Генерація зображень - це процес створення нових зображень на основі певних вхідних даних, у якості яких можу бути використано текстовий опис, набір параметрів або інше зображення.

Генерація зображень полягає в створенні нових зображень за допомогою використання алгоритмів генеративного навчання. Це включає в себе створення зображень, які виглядають так, ніби вони були створені людиною, але, насправді, вони були згенеровані алгоритмами штучного інтелекту.

Генерація зображень має широкий спектр застосувань, включаючи графічний дизайн, розробку ігор, медицинську діагностику та наукову симуляцію, а також створення нових форм мистецтва.

Однією з найбільш важливих задач генеративного мистецтва є досягнення високої якості вихідних даних із використанням мінімальної кількості часу та обчислювальної потужності на їх створення [1].

Основною перевагою використання генеративних моделей є здатність створювати зображення високої якості з мінімальними артефактами та великою реалістичністю. Це робить використання генеративних моделей важливим та ефективним інструментом для завдань, де важлива точність і відтворюваність даних, таких як медична діагностика, обробка зображень та симуляція.

У даній роботі буде розглянуто генеративні моделі, засновані на дифузії, а саме, їх специфічну гілку – моделі латентної дифузії, які стали нащадками

класичних дифузійних моделей, отримавши нову архітектуру та підход до реалізації процесу їх навчання та використання, що дозволило зробити їх використання більш оптимальним з точки зору їх технічних вимог до апаратного забезпечення обчислювальної машини, та зробило їх одними із наймасовіших генеративних моделей. Наразі моделі латентної дифузії є одними з найбільш використовуваних сучасних генеративних моделей, які отримала значний інтерес у наукових та технологічних колах. Моделі латентної дифузії мають широкий спектр застосувань. Вони можуть використовуватися для генерації нових зображень, відео та тексту. Вони також можуть використовуватися для регенерації зображень, які були пошкоджені або розмиті. Крім того, вони можуть використовуватися для створення нових творчих форматів, таких як колажі, гіперреалістичні зображення та віртуальна реальність.

Використання даного типу генеративних моделей дозволяє створення високоякісних та реалістичних даних, які базуються на наборі даних, на якому було навчено модель, проте генеровані дані є цілком новими. Основна ідея полягає в тому, щоб почати процес генерації з матриці випадкового шуму та поступово покращувати якість зображення, очищуючи шум у процесі генерації.

Використання моделей дифузії стало активно досліджуваною галуззю в сучасних дослідженнях, і вона продовжує розвиватися та розширювати свої можливості. Вона вже знайшла застосування в багатьох областях, і її роль у генерації зображень стає все більш важливою в сучасному світі комп'ютерної графіки і обчислювальної науки. Моделі латентної дифузії є потужним інструментом, який може бути використаний для створення нових і інноваційних даних. Даний клас генеративної моделі ще перебуває у розробці, але вони мають великий потенціал для зміни способу, яким людство створює дані.

1.1.2. Історичний огляд розвитку генерації зображень

Історичний контекст розвитку генерації зображень є ключовим для розуміння еволюції цієї динамічної галузі. За понад півстоліття існування,

генерація зображень виявила безліч переворотів та трансформацій, від технічних до концептуальних.

Початки генерації зображень пов'язані з підходами до векторної та растрової графіки. У 1960-1970 роках народилися перші системи, спроможні створювати прості растрові зображення, такі як лінії та форми. Це відкрило двері для застосування графіки у комп'ютерному середовищі, але якість та складність зображень були значно обмеженими.

У 1980-1990 роках з'явилися перші програми для рендеру тривимірних сцен, що відкрило шлях до створення більш складних та реалістичних зображень. Цей період також позначився народженням растрових графічних форматів, таких як BMP і TIFF, що стали стандартами для зберігання зображень.

Важливим кроком у розвитку генерації зображень був запуск досліджень з використанням генеративних змагальних мереж (GANs) у 2010 роках. GANs вперше запропонував Ян Гудфеллоу та його колеги. Ця ідея базується на концепції двох мереж: генератора, який створює зображення, і дискримінатора, який намагається визначити, чи є зображення реалістичним. Змагання між ними призводить до покращення якості створених зображень [2].

У цьому десятиріччі генерація зображень отримала новий поштовх завдяки глибокому навчанню. Завдяки нейронним мережам та архітектурі, такій як глибокі згорткові мережі (CNN) та рекурентні нейронні мережі (RNN), стало можливим створювати більш складні зображення. Також народилися методи згорткових генеративних змагальних мереж (cGANs) та варіаційних автоенкодерів (VAEs), що покращили якість та різноманітність зображень.

Новим етапом в генерації зображень є поява моделей, заснованих на дифузії, які стали об'єктом дослідження даної кваліфікаційної роботи. Використання моделей, заснованих на дифузії поєднує в собі ідеї з попередніх методів та вводить нові концепції, що дозволяють створювати високоякісні зображення з мінімальними артефактами.

Починаючи з 2020 року галузь генерації зображень залишається однією з найбільш активних інноваційних галузей обчислювальної графіки та машинного

навчання. Історія розвитку генерації зображень показує, які виклики та досягнення зумовили її поточний стан та перспективи подальшого росту.

1.1.3. Аналіз ключових досягнень у галузі генерації зображень

Розглядаючи історію генерації зображень, неможливо обійти увагою вражаючі досягнення, які визначили цю галузь. Даний підпункт присвячений аналізу ключових досягнень, успішних застосувань та видатних проектів у галузі генерації зображень.

Успіх генеративних змагальних мереж (GANs). Генеративні змагальні мережі (GANs) є одним із найважливіших досягнень у галузі. Вони призвели до створення захоплюючих зображень, які майже не відрізняються від реальних. Один із вражаючих прикладів успішного використання GANs - це глибокий перенос стилю (Deep Style Transfer), що дозволяє застосовувати художні стилі до зображень і фотографій [3].

Одним із останніх досягнень у галузі є моделі, засновані на дифузії. Вони розроблені для подолання проблем артефактів і забезпечення стабільної якості зображень. Проект Stable Diffusion від Stability AI відкрив нові можливості у генерації зображень та відкрив нові горизонти для застосування штучного інтелекту у галузі мистецтва, дизайну та фотографії.

Генерація контенту та відео стала важливим аспектом у галузі розважальної індустрії. Проекти, які використовують генерацію зображень для створення контенту, можуть автоматизувати процес створення анімацій, відеороликів та ігор. Одним із найвідоміших прикладів є робота DeepDream від Google, яка використовує нейронні мережі для створення глибокого мистецтва та психоделічних зображень [4]. Наразі компанія Stability AI також працює над розробкою моделей та алгоритмів, які змогли б дозволити генерацію якісних відеороликів та анімації.

Галузі медицини та науки також скористалися можливостями генеративних моделей. Відображення та аналіз медичних зображень, таких як

знімки МРТ та комп'ютерної томографії, стали точнішими завдяки застосуванню систем штучного інтелекту. Це дозволяє лікарям та науковцям отримувати більше інформації з зображень та поліпшувати діагностику та дослідження. Наприклад, ключовий компонент у архітектурі моделей латентної дифузії – нейронна модель U-Net, була вперше запропонована у 2015 році, як інноваційне рішення у сфері сегментації медичних зображень, через свою архітектуру обробки та сегментації вхідних даних. Однак, дана модель також знайшла ефективне використання у сфері генеративного мистецтва.

Успіхи в галузі генерації зображень не зупиняються, і майбутнє обіцяє ще більше інновацій. Розвиток глибокого навчання, вивчення архітектур, що поєднуються із розумінням людського сприймання, та розвиток обчислювальних ресурсів розширюють горизонти можливостей у галузі генерації зображень [5].

Загалом, аналіз ключових досягнень та успішних застосувань у галузі генерації зображень демонструє різноманітність та значущість цієї області для різних сфер життя і діяльності. Її розвиток змінює те, як ми створюємо, сприймаємо та використовуємо зображення у сучасному світі.

1.2. Сучасні підходи до генерації даних

1.2.1. Генеративні змагальні мережі (GAN)

У світі штучного інтелекту та глибокого навчання одним з найпримітніших досягнень є генеративні змагальні мережі (GAN), які створені для генерації контенту, що відповідає людському рівню якості. GAN є фундаментальним інструментом, що лежить в основі генерації зображень, і вони знаходять широке застосування у багатьох галузях, включаючи мистецтво із застосуванням комп’ютерного зору, медицину, дизайн та багато інших.

Генеративні змагальні мережі (GAN) - це клас нейронних мереж, які були вперше представлені Яном Лекуном і Іаном Гудфеллоу в 2014 році. Однак їх популярність вибухнула, коли було представлено додаткові роботи та

покращення в галузі глибокого навчання. Основна ідея полягає в тому, щоб навчити дві нейронні мережі одночасно: генератор та дискримінатор.

Генератор (Generator) призначений для створення даних, таких як зображення, звуки, текст тощо. Він приймає на вході випадковий шум та намагається створити дані, які є найбільш схожими на реальні. Генератор поступово навчається підбирати оптимальні параметри для виходу, які близькі до реальних даних.

Дискримінатор (Discriminator), навпаки, призначений для відрізнення справжніх даних від тих, що згенеровані генератором. Його завдання - фільтрувати "підроблені" дані та визначити, наскільки вони подібні до реальних. Дискримінатор також навчається, оптимізуючи параметри для кращого розрізнення між реальними та згенерованими даними [6].

Принцип роботи GAN полягає в тому, що генератор та дискримінатор грають "гру". Генератор намагається виграти цю гру, створюючи дані, які неможливо відрізнити від реальних, тим часом як дискримінатор намагається виявити підроблені дані. Ця взаємодія веде до поступового покращення якості генерованих даних. Принцип роботи змагальних мереж наведено на рис. 1.1.

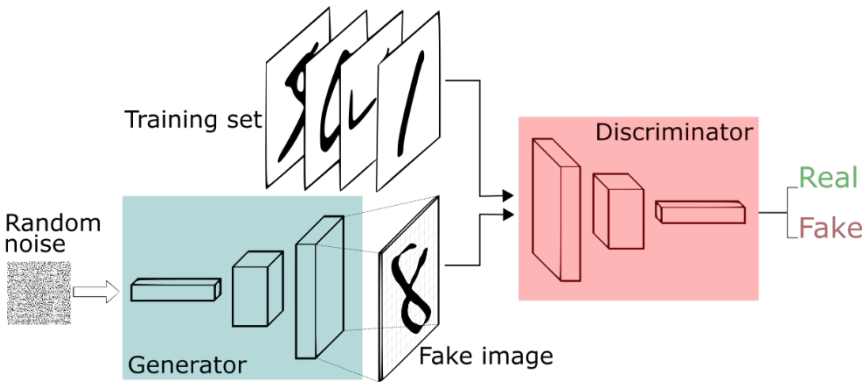

Рис. 1.1. Принцип роботи змагальних мереж

Генеративні змагальні мережі мають численні переваги, включаючи:

– складність генерованих даних: GAN здатні генерувати складні дані, такі як високоякісні зображення та аудіофайли, що робить їх корисними в галузях мистецтва та дизайну;

– застосування в медицині: GAN можуть використовуватися для генерації зображень медичних даних, таких як рентгенівські знімки або зображення МРТ. Це може допомогти лікарям в точній діагностиці та лікуванні;

– автоматизація дизайну: У сфері дизайну GAN можуть створювати унікальні малюнки, шаблони та концепції, які можуть бути використані для створення нових продуктів;

– покращення генерації зображень: GAN використовуються для покращення якості та реалізму генерованих зображень;

– глибоке навчання: GAN є однією з ключових моделей в глибокому навчанні та представляють важливий напрямок для подальшого дослідження.

Отже, генеративні змагальні мережі є надзвичайно потужним інструментом для генерації зображень та інших типів даних. Їх застосування в різних галузях надає можливість створювати новий контент, покращувати процеси діагностики та автоматизації дизайну. На сьогодні GAN є однією з ключових технологій у світі штучного інтелекту та глибокого навчання, і вони продовжують розвиватися, відкриваючи нові можливості для інновацій та мистецтва.

Зрозуміло, що генеративні змагальні мережі не обмежуються реалізмом відображення. Вони відкривають двері до створення мистецтва, яке неможливо було уявити раніше, та дозволяють нам використовувати штучний інтелект для створення мистецтва у нових формах та представленнях. У галузі генеративного мистецтва, також званого "AI Art," митці використовують GAN для створення унікальних, сюрреалістичних, інноваційних та деколи вражаючих шедеврів.

Генеративні змагальні мережі - це потужний інструмент, який змінює підхід до створення зображень та даних взагалі. Вони відкривають можливості для створення мистецтва, автоматизації, діагностики та багатьох інших галузей. Завдяки їхній унікальній здатності "навчатися" і покращуватися з кожним поколінням, люди можуть бути свідками неймовірного прогресу в цьому напрямку.

1.2.2. Моделі, засновані на дифузії

У сучасному світі генерація зображень стала значущим напрямком розвитку, особливо в контексті штучного інтелекту та машинного навчання. Одним із інноваційних методів, які зробили великий прорив в даній галузі, є метод генерації зображень за допомогою дифузії. Це метод, який базується на глибокому навчанні та нейромережах, відкриває нові можливості у сфері створення високоякісних зображень та мистецтва [7].

1.2.2.1. Принцип роботи моделей дифузії

Моделі дифузії - це тип генеративної моделі, які використовуються для створення зображень з випадкових шумів. Модель працює шляхом послідовного зменшення кількості шуму у вхідному зображенні, при цьому зберігаючи основні характеристики зображення. Процес навчання таких моделей схожий на процес дифузії в фізиці, коли молекули поступово перемішуються і рівномірно розподіляються в просторі, це є гарною аналогією до того як модель під час навчання додає шум до вхідного зображення для його подальшої оцінки, видалення та формування вихідного зображення. Процес генерації же правильно буде описати як обернену дифузію (reversal diffusion), коли зашумлене вхідне зображення покроково очищується від шуму, формуючи вихідне зображення.

Розглянемо фундаментальні принципи роботи моделей стабільної дифузії. Модель стабільної дифузії навчається на наборі даних зображень, цей набір даних може включати в себе різні типи зображень, такі як фотографії, картини, ілюстрації тощо. Процес генерації зображень починається з того, що модель генерує початкове представлення зображення у латентному просторі, яке є набором випадкових шумів. Після цього модель поступово зменшує кількість шуму в латентному представленні зображення, при цьому зберігаючи основні характеристики зображення відповідно до текстового опису. Процес видалення шуму зупиняється, коли зображення досягає бажаної якості.

Процес зменшення шуму в моделі стабільної дифузії здійснюється за допомогою спеціального алгоритму – методу вибірки, який реалізує процес зворотної дифузії. Приклад роботи дифузійної моделі дифузії наведено на рис. 1.2.

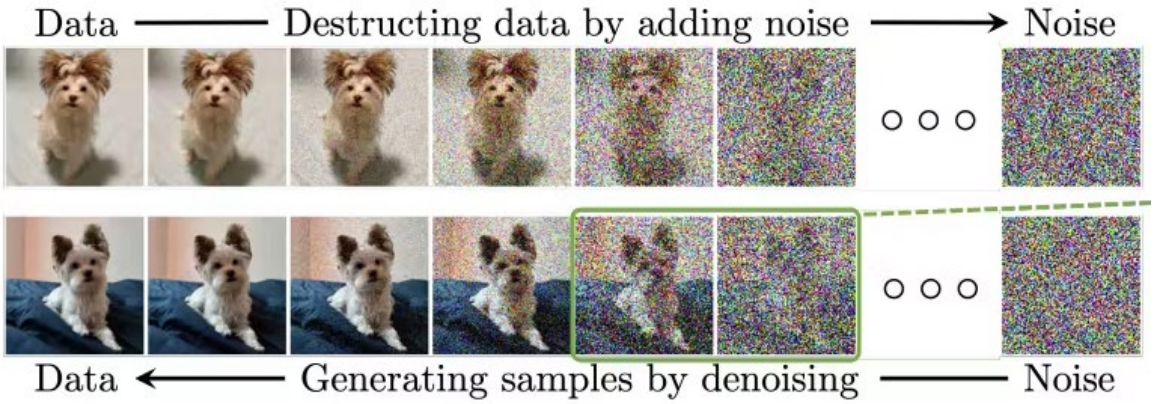

Рис. 1.2. Приклад роботи моделі дифузії

Дифузійна модель має ряд переваг перед іншими типами генеративних моделей. По-перше, вона може генерувати зображення високої якості, які важко відрізнити від реальних фотографій. По-друге, вона може генерувати зображення з різними стилями, такими як реалістичний, абстрактний, художній тощо. По-третє, вона може генерувати зображення за заданими текстовими інструкціями.

Модель стабільної дифузії має широкий спектр можливостей застосування. Вона може використовуватися для створення зображень для реклами, дизайну, освіти тощо. Також вона може використовуватися для створення нового творчого контенту, такого як мистецтво, музика, література тощо.

1.2.2.2. Особливості та переваги дифузійних моделей

Використання моделей стабільної дифузії має певні переваги у порівнянні із іншими існуючими підходами у сфері генерації зображень:

– контроль якості зображень. Однією з великих переваг моделей дифузії є можливість контролювати якість та чіткість зображень. Під час процесу дифузії можна налаштовувати параметри для досягнення бажаного результату, що робить цей метод особливо корисним у вимірюванні якості зображень;

– висока різноманітність зображень. Дифузійні моделі дозволяють створювати різноманітні зображення від абстрактних малюнків до фотореалістичних фотографій;

– навчання за меншим обсягом даних. Моделі дифузії можуть бути навчені за меншою кількістю даних порівняно з іншими методами генерації зображень. Це особливо важливо в випадках, коли обмежено доступ до великих обсягів даних, також це робить навчання дешевшим.

1.2.2.3. Застосування моделей дифузії

Наразі застосування моделей, заснованих на дифузії, знаходить місце у багатьох професійних сферах, наприклад:

– графічний дизайн. У графічному дизайні моделі дифузії можна використовувати для створення реалістичних зображень для реклами, упаковки, веб-дизайну та інших цілей. Моделі дифузії можуть генерувати зображення високої якості, які важко відрізнити від реальних фотографій. Це робить їх цінним інструментом для графічних дизайнерів, які хочуть створити професійні та реалістичні зображення;

– мистецтво. Моделі дифузії можна використовувати для створення нових форм мистецтва, таких як генеративне мистецтво. Генеративне мистецтво - це тип мистецтва, який створюється за допомогою алгоритмів машинного навчання. Моделі дифузії можуть використовуватися для створення нових і несподіваних форм мистецтва, які неможливо було б створити вручну. Моделі дифузії відкривають нові можливості для створення мистецтва. Вони можуть використовуватися для створення нових і інноваційних форм мистецтва, які неможливо було б створити вручну;

– відеоігри. Моделі дифузії можна використовувати для створення реалістичних персонажів, фонів і об'єктів для відеоігор. Наприклад, їх можна використовувати для створення реалістичних персонажів з деталізованою зовнішністю та рухами, реалістичних фонів, які створюють відчуття занурення в гру, реалістичних об'єктів, які додають грі реалістичності;

– медицина. Моделі дифузії можна використовувати для створення симуляцій медичних процедур, таких як хірургія або рентген. Наприклад, їх можна використовувати для створення симуляцій хірургічних процедур, які можуть використовуватися для навчання лікарів або симуляцій рентгенівських знімків, які можуть використовуватися для діагностики захворювань;

– наука. Моделі дифузії можна використовувати для створення зображень з наукових даних, таких як клітинні структури або космічні об'єкти. Наприклад, їх можна використовувати для створення зображень клітинних структур, які можуть використовуватися для вивчення біології або зображень космічних об'єктів, які можуть використовуватися для вивчення астрономії. Моделі дифузії можуть допомогти науковцям краще зрозуміти складні наукові дані.

Це лише деякі приклади того, як моделі дифузії можна використовувати в різних сферах. Застосування моделей дифузії продовжує розвиватися, і, ймовірно, їх можна буде використовувати в нових сферах у майбутньому.

1.2.3. Трансформери (Transformers)

Трансформери, спочатку розроблені для завдань обробки природної мови, швидко стали популярним інструментом у сфері генерації зображень завдяки їх універсальності та здатності пристосовуватися до різних завдань. Трансформери - це архітектура нейромережі, яка базується на механізмах уваги (attention та self-attention). Вони складаються з декількох повторюваних шарів, відомих як «трансформерні блоки». Кожен такий блок містить дві основні компоненти: механізм уваги та позиційну зв’язність. Механізм уваги дозволяє

моделі взаємодіяти з різними частинами вхідних даних та приділяти їм вагу в залежності від контексту, що робить трансформери дуже потужними для моделювання складних залежностей у даних. Позиційна зв’язність враховує порядок та розташування даних у вхідній послідовності.

Основна ідея роботи трансформерів полягає в тому, що вони обробляють вхідні дані паралельно та незалежно. Кожен блок трансформера приймає вхідні дані та генерує вихід, що передається наступному блоку. Цей процес повторюється кілька разів (зазвичай декілька десятків блоків), що дозволяє моделі адаптуватися до різних аспектів вхідних даних.

Трансформери швидко завоювали популярність у завданнях генерації зображень. Однією з найпоширеніших застосувань є моделі для автоматичного підпису зображень та генерації мистецтва. Такі моделі можуть аналізувати вміст зображення та створювати відповідні описи або ж зображення на основі текстового опису. Трансформери також застосовуються у сферах графічного дизайну, обробки зображень, створення спеціальних ефектів та багатьох інших областях.

Однією з сильних сторін трансформерів є їх здатність використовувати передбачувані прийоми навчання та генерації. Такий підхід дозволяє керувати процесом генерації та створювати бажані характеристики зображень [8].

Трансформери є потужним інструментом у сфері генерації зображень завдяки своїй універсальності та здатності до застосування в різних галузях. Вони відкривають нові можливості для творчості та мистецтва та надають можливість впливати на процес генерації зображень, що робить їх надзвичайно цікавими та потужними інструментами у мистецтві та наукових дослідженнях.

1.2.4. Варіаційні автоенкодери (VAE)

Варіаційні автоенкодери (VAE) представляють собою потужний клас моделей у глибинному навчанні, які володіють значними перевагами у сфері генерації зображень та репрезентації даних.

VAE базується на автоенкодерах - нейромережевих моделях, які використовуються для зменшення розмірності даних та створення їхньої латентної репрезентації. Основна ідея VAE полягає в тому, щоб перетворити вхідні дані в їх латентне представлення із розмірністю значно меншою за оригінальну, де кожен пункт у цьому просторі відповідає певному представленню. Важливо відзначити, що цей простір може бути визначений як розподіл ймовірностей, який дозволяє вимірювати невизначеність в утворенні латентних репрезентацій. Ця особливість робить VAE потужним інструментом для моделювання реальних даних, особливо в контексті генерації зображень [9].

Генерація зображень з VAE відбувається в декілька етапів. Спершу, зразок з латентного простору генерується за допомогою деякого розподілу ймовірностей, зазвичай, нормального розподілу. Потім цей зразок подається на вхід декодеру, який перетворює його в оригінальне зображення. Головна ідея полягає в тому, що VAE може генерувати зразки, які подібні до тих, що були використані для навчання. Оскільки латентний простір може бути розглянутий як простір розподілу ймовірностей, VAE може генерувати різноманітні зображення, зберігаючи важливі структурні особливості. Принцип роботи VAE наведено на рис. 1.3.

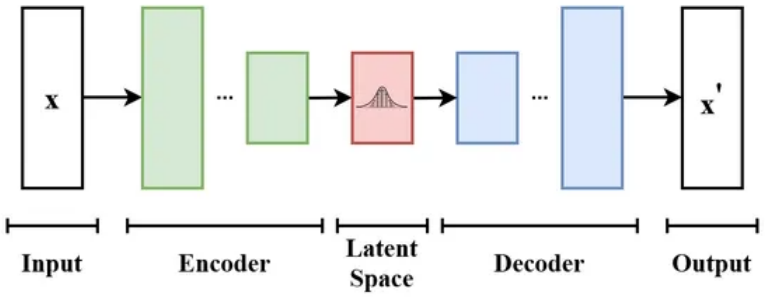

Рис. 1.3. Принцип роботи VAE

VAE мають безліч застосувань у галузі генеративного мистецтва. Вони можуть бути використані для створення художніх зображень, структуризації музики, генерації тексту, та навіть для комбінування різних видів мистецтва у творчій формі. Однією з ключових переваг VAE є їхня здатність до створення

різноманітних варіацій творів, що робить їх цінним інструментом для художників та творців.

1.3. Огляд популярних систем генерації зображень

Сучасні технології генеративного навчання дозволяють створювати реалістичні та високоякісні зображення з використанням штучного інтелекту. У даному пункті буде розглянуто деякі з найпопулярніших сервісів, які використовують моделі генеративного навчання для створення зображень, а також їхні основні характеристики, такі як переваги і недоліки.

1.3.1. Stable Diffusion

У даному контексті під назвою Stable Diffusion слід розуміти усі сервіси, які надають послуги з генерації зображень, та використовують моделі латентної дифузії Stable Diffusion, або моделі, що базуються на моделях Stable Diffusion. Stable Diffusion – це комплексний метод генеративної моделі, який використовує латентні дифузійні моделі для створення зображень. Модель Stable Diffusion v1.1 була вперше представлена колаборацією компаній CompVis, Stability AI та Runway у 2022 році. Ними ж було випущено серію моделей, які були покращеними версіями попередніх. Використання даної серії моделей швидко стало одним із найпопулярніших підходів до створення зображень. Основні характеристики:

– стабільність. Stable Diffusion є більш стабільним, ніж інші методи генеративної моделі. Це означає, що він менше схильний до збою або створення зображень низької якості;

– якість. Моделі Stable Diffusion генерують зображення вищої якості, ніж інші методи генеративної моделі;

– швидкість. Використання моделей латентної дифузії є досить швидким, це робить їх придатними для створення великих зображень або для використання в реальному часі [10];

– обчислювальна ефективність. Моделі латентної дифузії потребують значно меншої обчислювальної потужності, порівняно із іншими генеративними моделями.

Модель Stable Diffusion схожа на інші генеративні моделі, засновані на дифузії, однак даний клас моделей відрізняється своєю архітектурою, яка надає їй ряд значних переваг, таких як стабільність, якість вихідних зображень і швидкість генерації, які роблять її більш ефективним та універсальним рішенням у задачах генеративного мистецтва.

1.3.2. MidJourney

MidJourney використовує метод генеративної моделі, заснований на дифузії, для створення реалістичних і високоякісних зображень. MidJourney має широкий спектр функцій, включаючи можливість використання текстових запитів для створення зображень, які відповідають заданим вимогам.

Основні характеристики:

– реалізм: MidJourney генерує реалістичні зображення, які важко відрізнити від реальних фотографій;

– широкий спектр функцій. MidJourney має широкий спектр функцій, які дозволяють користувачам створювати різноманітні зображення;

– платна підписка. Для використання MidJourney у комерційних цілях потрібно придбати ліцензію [11].

MidJourney має ряд переваг, таких як реалізм і широкий спектр функцій, які є визначеними у налаштуваннях моделі, що спрощує процес генерації зображень для недосвідчених користувачів, проте є менш гнучким, якщо, наприклад, потрібно змінити стилістику або кольорову гаму зображення. Через це більшість зображень, що були згенеровані із використанням MidJourney мають схожу стилістику та не можуть похвастатися значною оригінальністю. До того ж дана модель є значно більш вибагливою до технічних характеристик обчислювальної машини.

1.3.3. DALL-E 2

DALL-E 2 - це система штучного інтелекту, розроблена OpenAI, яка революціонізувала сферу генерування зображень. Вона володіє надзвичайною здатністю створювати реалістичні та захоплюючі зображення з простих текстових описів, відкриваючи новий кордон творчості та уяви.

В основі майстерності DALL-E 2 лежить унікальне поєднання двох потужних моделей штучного інтелекту: моделі дифузії та моделі вбудовування зображень CLIP. Модель дифузії діє як художник, ретельно створюючи зображення піксель за пікселем, поступово вдосконалюючи його від шумного стану до впізнаваної форми. Модель вбудовування зображень CLIP, з іншого боку, відіграє роль арт-критика, надаючи керівництво та зворотний зв'язок моделі дифузії, гарантуючи, що згенероване зображення відповідає текстовому опису та відображає задуману користувачем суть.

Можливості DALL-E 2 виходять далеко за межі простого генерування фотореалістичних зображень. Він також може створювати абстрактні візерунки, сюрреалістичні пейзажі та навіть ілюстрації антропоморфних персонажів. Також даний сервіс може модифікувати існуючі зображення, додаючи або видаляючи елементи, зберігаючи при цьому узгодженість із загальною сценою.

Застосування DALL-E 2 настільки ж різноманітні, як і його творчий потенціал. Він може використовуватися в дизайні продуктів для генерування прототипів та візуалізації інноваційних концепцій. Він може допомагати в освіті, створюючи захоплюючі ілюстрації та діаграми для підвищення ефективності навчання. Він навіть може служити інструментом для особистого самовираження, дозволяючи людям втілити свою уяву в життя через яскраві образи. DALL-E 2 представляє значний стрибок вперед у технології штучного інтелекту, демонструючи свою здатність не тільки обробляти інформацію, а й створювати та впроваджувати інновації [12].

1.3.4. DeepDream

DeepDream - це метод генеративного навчання, розроблений Google AI у 2015 році. DeepDream працює, аналізуючи зображення за допомогою нейронної мережі, яка навчена розпізнавати об'єкти та текстури. DeepDream використовує нейронну мережу для виявлення і підкреслення візуальних паттернів у зображеннях і на основі цих даних генерує нове зображення на базі вхідного. DeepDream був вперше представлений на конференції Google I/O 2015 року і швидко став популярним у Інтернеті. Він був використаний для створення ряду вражаючих і химерних зображень, які були опубліковані в соціальних мережах та блогах.

DeepDream працює, надаючи зображення згортковій нейронній мережі, яка була навчена на величезній базі даних зображень. Мережа шукає візерунки в зображенні, які відповідають шаблонам, які вона навчилася розпізнавати. Потім вона посилює ці візерунки, додаючи шум або інші зміни до зображення. Цей процес повторюється кілька разів, поки зображення не набуде бажаного вигляду.

DeepDream є прикладом того, як нейронні мережі можуть використовуватися для створення творчих та оригінальних зображень. Він також є прикладом того, як нейронні мережі можуть бути використані для виявлення візерунків у реальному світі. DeepDream може бути використана для створення візуальних ефектів. DeepDream може використовуватися для створення різних візуальних ефектів, таких як галюцинації, візерунки та текстури.

DeepDream відрізняється від інших методів генеративної моделі тим, що він не створює нові зображення з нуля, а лише змінює існуючі зображення. Це дозволяє йому створювати більш реалістичні зображення, які важко відрізнити від реальних фотографій.

1.3.5. Nvidia GauGAN

NVIDIA GauGAN - це генеративна нейронна мережа, розроблена компанією NVIDIA. Вона використовує метод машинного навчання, який

називається трансферним навчанням, щоб навчитися генерувати реалістичні зображення з простих ескізів. GauGAN була представлена на конференції SIGGRAPH 2018 року і швидко стала популярною. Вона була використана для створення ряду вражаючих зображень, які були опубліковані в соціальних мережах та блогах.

GauGAN працює, надаючи простий ескіз згортковій нейронній мережі, яка була навчена на великій базі даних зображень. Мережа використовує ескіз як вказівку для створення реалістичного зображення. Вона робить це, використовуючи свої знання про те, як виглядають різні об'єкти та візерунки.

GauGAN може генерувати реалістичні зображення різних об'єктів та сцен, таких як пейзажі, портрети та архітектура. Вона також може генерувати зображення з різними стилями, такими як реалістичний, абстрактний або навіть карикатурний. GauGAN є прикладом того, як нейронні мережі можуть використовуватися для створення творчих та оригінальних зображень. Вона також є прикладом того, як нейронні мережі можуть бути використані для автоматизації завдань, які раніше виконувалися людьми. Основні характеристики Nvidia GauGAN:

– реалізм: GauGAN генерує реалістичні зображення, які важко відрізнити від реальних фотографій;

– широкий спектр функцій: GauGAN має широкий спектр функцій, які дозволяють користувачам створювати різноманітні зображення;

– доступність: GauGAN доступний у бета-версії для обмеженої кількості користувачів;

– платна підписка. Для використання GauGAN у комерційних цілях потрібно придбати ліцензію [14].

Отже, підводячи підсумки, можна звести усі наведені дані у вигляді наступної таблиці, що містить назву сервісу, метод, на якому базується генеративна модель, що використовується під час генерування та основні характеристики-переваги кожного з сервісів наведено у таблиці 1.1.

Таблиця 1.1.

Порівняння систем генерації зображень

Система	Тип моделі	Основні характеристики
Stable Diffusion	Модель латентної дифузії	Стабільність, якість, швидкість, можливість використання текстових запитів
MidJourney	Дифузійна модель	Реалізм, широкий спектр функцій, можливість використання текстових запитів
DALL-E 2	Дифузійна модель	Можливість створення різних творчих форматів, доступність до широкого набору даних, можливість використання текстових запитів
DeepDream	Згорткова нейронна мережа	Створення візуальних ефектів
Nvidia GauGAN	GAN	Реалізм, широкий спектр функцій, можливість використання текстових запитів

Список існуючих систем та моделей генерації зображень не обмежується лише наведеними прикладами, однак, ті, що представлені у таблиці є прикладами популярних на даний момент систем генерації зображень, які мають значний попит та інтерес у користувачів та розробників. Можна помітити, що більшість популярних систем генерації зображень базуються на моделях дифузії, що лише підкреслює виняткову роль даного класу генеративних моделей та їх важливість у контексті розвитку генеративного мистецтва.

1.4. Проблематика вирішення завдань генеративного мистецтва

Сучасні підходи до виконання завдань генеративного мистецтва ґрунтуються на використанні нейронних мереж та алгоритмів машинного навчання. Нейронні мережі є потужними інструментами, які дозволяють генерувати реалістичні та творчі зображення. Однак вони мають низку обмежень, які впливають на їхню продуктивність та ефективність.

Одним із основних обмежень сучасних нейронних мереж є їхня потреба в великому обсязі даних для навчання. Для навчання нейронної мережі, яка здатна

генерувати реалістичні зображення, необхідно використовувати величезну кількість зображень, які є прикладами того, що мережа повинна генерувати. Це може бути складним завданням, оскільки може бути важко зібрати достатньо зображень високої якості.

Іншим обмеженням сучасних нейронних мереж є їхня потреба в потужних обчислювальних ресурсах. Нейронні мережі є складними математичними моделями, які потребують значних обчислювальних ресурсів для навчання та використання. Це може бути проблемою для користувачів, які мають доступ до обмежених обчислювальних ресурсів.

Таким чином, високі вимоги до обчислювальних ресурсів ЕОМ є ключовим обмеженням, яке робить навчання та використання генеративних моделей занадто коштовним та недоступним для масового використання.

Вирішенням даної проблеми є підвищення ефективності архітектури та алгоритмів, що задіяні у генеративних процесах. Таким чином, моделі латентної дифузії через свою нову архітектуру стали інноваційним рішенням, яке дозволило створення даних високої якості при цьому суттєво зменшивши вимоги до обчислювальної потужності ЕОМ, що зробило навчання та використання даного класу моделей значно ефективнішим у задачах генеративного мистецтва.

1.5. Висновок до першого розділу

У першому розділі було розглянуто основи генерації зображень та проведено аналіз поточних засобів та інструментів, що використовуються у задачах генеративного мистецтва. Вивчаючи історичний контекст та історичний огляд розвитку генерації зображень, було виявлено, що ця галузь пройшла вражаючий шлях від перших спроб генерації зображень до сучасних досягнень та успішних застосувань. Серед основних підходів до створення зображень було проаналізовано використання наступних типів генеративних моделей:

- генеративні змагальні мережі (GAN);
- моделі, засновані на дифузії;

– моделі, засновані на латентній дифузії;

– трансформери (Transformers);

– варіаційні автоенкодери (VAE).

Можна побачити, що використання моделей, заснованих на дифузії, наразі є одним із найбільш вживаних підходів до вирішення завдань генеративного мистецтва. Однак, класичні дифузійні моделі, як і інші наведені генеративні моделі, мають суттєвий недолік – їх високі вимоги до обчислювальних характеристик ЕОМ, які задіяні під час обчислення та зберігання даних, що використовуються у процесі генерації. Через це було обрано похідний клас від моделей дифузії – моделі латентної дифузії. Даний клас моделей архітектурно відрізняється від класичних дифузійних моделей через інноваційну архітектуру, впровадження якої дозволило зробити значний ривок у оптимізації швидкості та вартості їх навчання та використання. Це зробило використання моделей латентної дифузії доступним для широкого кола користувачів, які зацікавлені у дослідженні та використанні моделей машинного навчання у контексті генеративного мистецтва.

Подальші дослідження будуть присвячені теоретичному та практичному дослідженню моделей латентної дифузії, їх алгоритмів навчання та використання у процесі генерації зображень, архітектури. Буде визначено призначення параметрів генерації та вплив їх значень на:

– якість вихідних зображень;

– відтворюваність зображень;

– швидкість процесу генерації;

– керування процесом генерації;

– ефективність використання обчислювальних ресурсів.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ МОДЕЛЕЙ ЛАТЕНТНОЇ ДИФУЗІЇ

2.1. Моделі, засновані на дифузії

Моделі, засновані на дифузії, - це тип генеративних моделей, які використовують процес дифузії для генерації зображень. Процес дифузії починається з випадкового шуму і поступово розмиває шум, поки він не перетвориться на зображення, яке схоже на одне з набору даних, на якому модель була навчена. Даний алгоритм генеративного навчання може бути використано для створення зображень, музики, тексту та інших форм творчого контенту [15].

Моделі дифузії можна поділити на два основних типи:

– моделі прямої дифузії починаються з випадкового зображення або тексту і поступово додають деталі, поки зображення або текст не набуде бажаного вигляду;

– моделі зворотної дифузії починаються з бажаного зображення або тексту і поступово видаляють деталі, поки зображення або текст не стане випадковим.

Моделі дифузії мають ряд переваг у порівнянні з іншими типами генеративних моделей. Вони можуть створювати зображення, які є більш реалістичними та деталізованими, ніж зображення, створені іншими моделями. Вони також можуть створювати зображення з різними стилями, такими як реалістичний, абстрактний або навіть карикатурний. Моделі дифузії можуть генерувати зображення з високою роздільною здатністю та вони відносно прості в реалізації. До того ж вони можуть бути використані для генерації різних типів зображень, включаючи реалістичні, творчі та абстрактні зображення. Однак, вони мають і суттєвий недолік – потреба у значній обчислювальній потужності, яка робила тренування та використання даних моделей недоступними для широкого кола користувачів [16].

Для вирішення даної проблеми розробниками із CompVis, StabilityAI та Runway було розроблено на представлено моделі латентної дифузії.

2.2. Латентні дифузійні моделі

Створення моделей латентної дифузії було наступним значним кроком у розвитку генеративних моделей, заснованих на дифузії. Перша офіційна версія моделі Stable Diffusion, відома як Stable Diffusion v 1.1, була представлена компаніями CompVis, Stability AI та Runway у 2022 році на конференції NeurIPS. Модель отримала позитивні відгуки від наукової спільноти та була відзначена як значне досягнення у галузі розвитку генеративних моделей [17].

Ключовою перевагою нової архітектури моделі є перенос обчислень під час навчання та генерації зображень з піксельного простору (робота з пікселями вхідного зображення) у латентний простір (latent space), тобто дані моделі працюють із числовим представленням вхідного зображення у більш низькій розмірності, а не самим зображенням. Такий підхід дозволив значно скоротити час та витрати на навчання даних моделей, а також аналогічно пришвидшити та здешевити процес генерації зображень із використанням даних моделей. Щоб створити зображення на основі текстового опису, модель латентної дифузії починає з випадкового шуму. Потім модель поступово «розмиває» шум, формуючи зображення, яке відповідає текстовому опису.

Ключовою перевагою даного архітектурного рішення є те, що генерація виконується шляхом попереднього стискання зображення в латентний простір нижчого виміру, замість того щоб виконувати обробку у піксельному просторі оригінальної розмірності, це є ефективнішим способом подання зображення. Це дозволяє моделі генерувати зображення, які більш узгоджуються з текстовим описом і менш ймовірно містять артефакти або помилки. Також це дозволило значно зменшити апаратне навантаження, час на обробку та вимоги до апаратної складової обчислювальної машини, адже латентний простір являє собою зжате інформаційне представлення зображення, яке значно менше за оригінальне зображення (його піксельний простір). Наприклад, зображення розміром 512x512 пікселів, яке при використанні звичайної дифузійної моделі виглядало би як матриця (3,512,512), яка відображає кожен з трьох кольорових каналів

зображення, висоту та ширину, мала б загальну кількість параметрів для обчислення (3x512x512) = 786432. Латентне представлення того ж зображення буде складати (4x64x64), що відповідає 16384 параметрам, які необхідно обчислювати та зберігати у пам’яті, тобто у 48 разів менше. Таким чином, робота із латентним представленням зображення є надзвичайно ресурсо-орієнтованим рішенням, яке революціонізувало підхід до обчислення та збереження параметрів під час процесу генерації зображення [18].

Латентний простір - це вектор чисел, який представляє основні особливості зображення. Наприклад, латентний простір може включати такі особливості, як наявність об'єктів, кольори об'єктів і просторові взаємозв'язки між об'єктами. Шляхом стиснення зображення в латентний простір Stable Diffusion може зосередитися на цих основних особливостях і ігнорувати деталі, які не важливі для текстового опису (Prompt).

Після того, як зображення було стиснуто в латентний простір, Stable Diffusion може почати генерувати нові зображення. Вона робить це шляхом поступового додавання шуму до латентного представлення зображення. Спочатку зображення буде дуже шумним і важко розрізненим. Однак у міру того, як модель буде прогнозувати та видаляти шум із представлення, зображення поступово ставатиме чіткішим і більш впізнаваним.

Останнє зображення, яке генерується Stable Diffusion, є поєднанням текстового опису та латентного представлення зображення. Текстовий опис дає моделі загальне уявлення про те, як має виглядати зображення, а латентне представлення забезпечує моделі конкретними деталями, які роблять зображення унікальним [19].

2.3. Архітектура моделей латентної дифузії

Модель стабільної дифузії не є монолітною, а складається із набору кількох компонентів та моделей:

– варіаційного автоенкодер (VAE);

– моделі U-Net;

– трансформеної моделі CLIP.

Схему взаємодії VAE, U-Net та CLIP у складі моделі стабільної дифузії під час генерації зображення наведено на рис. 2.1.

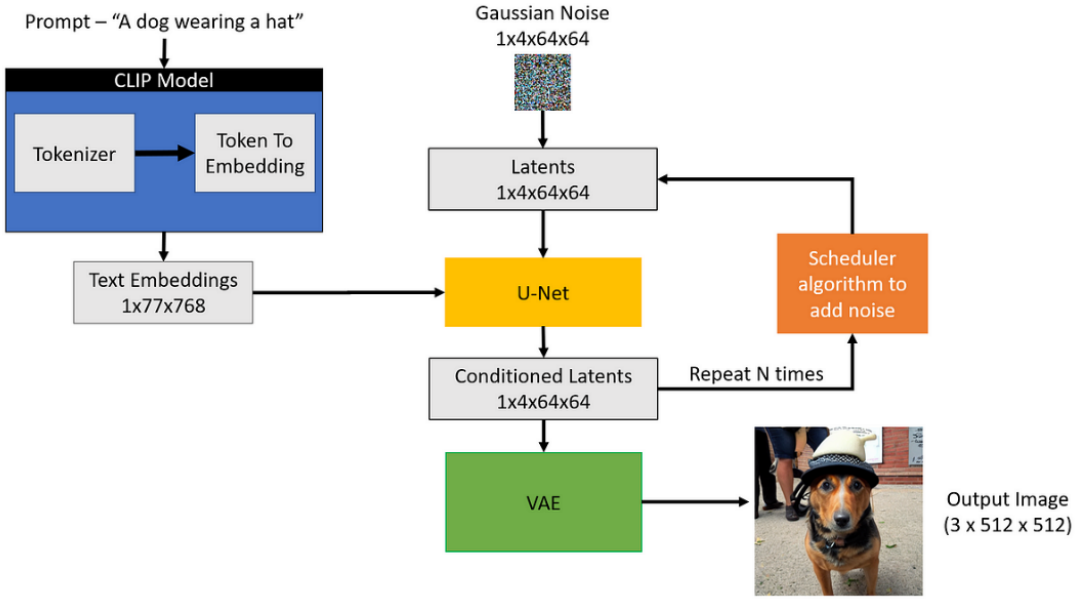

Рис. 2.1. Схема взаємодії VAE, U-Net та CLIP під генерації зображення

Підсумовуючи наведено схему. Робочий процес починається з генерації вектору початкового шуму (Gaussian noise) на основі параметра seed. Далі текстовий запит користувача кодується за допомогою енкодера тексту CLIP. Це кодування виконується для перетворення вхідного текстового опису на вектор, який буде додано до латентного простору під час генерації, це дозволить моделі U-Net слідувати вказівкам із запиту. U-Net повторює кроки обробки: прогноз шуму у представленні зображення у латентному просторі та видалення частки шуму. Робочий процес повторюється N разів за допомогою алгоритму планування. Вони виконуються задану кількість разів доки шум не буде видалено. Після того як U-Net закінчив видалення шуму, декодер VAE перетворює матрицю латентного шуму у зображення високої роздільної здатності. Остаточним результатом робочого процесу є зображення, яке відповідає текстовому запиту користувача [20]. Тепер розглянемо архітектуру кожного з компонентів окремо.

2.3.1. Варіаційні автоенкодери (VAE)

У контексті Stable Diffusion, варіаційний автокодер (VAE) є нейронною мережею, яка відіграє вирішальну роль у створенні високоякісних зображень із текстових описів. Вона служить мостом між оригінальним зображенням та його стисненим представленням у латентному просторі.

VAE складається з двох основних компонентів:

– Енкодер діє як компресор, приймаючи зображення як вхід і перетворюючи його в представлення нижчого виміру в латентному просторі. Цей латентний простір захоплює основні риси зображення, відкидаючи нерелевантні деталі.

– Декодер виконує протилежне завдання, відновлюючи зображення з латентного представлення. Він перетворює числове представлення зображення назад у піксельне зображення [21]. Схему роботи варіаційного автоенкодера VAE наведено на рис. 2.2.

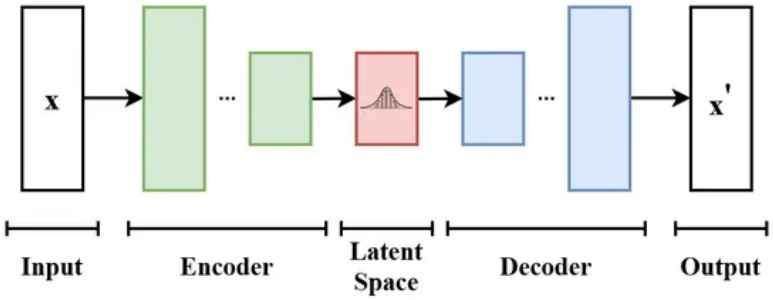

Рис. 2.2. Схема роботи варіаційного автоенкодера

Латентний простір, будучи значно меншим, ніж оригінальний піксельний простір зображення, надає кілька переваг:

– обчислювальна ефективність: робота в латентному просторі значно зменшує обчислювальні витрати, дозволяючи прискорити обробку та навчання;

– стійкість до шуму: представлення латентного простору менш сприйнятливе до шуму порівняно з піксельним представленням, що призводить до чіткішого та послідовнішого створення зображень;

– вилучення ознак: латентний простір ефективно захоплює ключові характеристики зображення, полегшуючи моделі навчання та інтерпретацію вмісту зображення.

Під час навчання замість того, щоб додавати шум безпосередньо до зображення, модель латентної дифузії вводить шум у латентне представлення зображення. Цей підхід виявляється більш ефективним через зменшену розмірність латентного простору.

По суті, VAE відіграє центральну роль у оптимізації процесу навчання та створення зображень моделями латентної стабільної дифузії, дозволяючи їм створювати високоякісні, деталізовані та семантично значущі зображення з текстових описів при значно менших витратах часу та обчислювальної потужності аніж при інших методах генерації зображень [22].

2.3.2. Модель U-Net

U-Net - це архітектура згорткової нейронної мережі (CNN), спеціально розроблена для завдань сегментації та генерації зображень. Вона стала популярним вибором для цих завдань завдяки здатності точно захоплювати та відтворювати ознаки зображень, навіть при низькій роздільній здатності вхідного зображення або зашумлених вхідних даних. U-Net є одним з ключових компонентів, адже у складі модель стабільної дифузії дана мережа під час навчання тренується розпізнавати деталі у латентному представленні зображення та оцінювати шум, доданий до зображення. Архітектура U-Net має структуру енкодер-декодер [23].

Енкодер відповідальний за процес видобування високорівневих ознак з вхідного зображення, шлях енкодера складається з серії згорткових та агрегувальних шарів, які поступово зменшують роздільну здатність зображення та збільшують кількість каналів ознак. Цей процес ефективно стискає інформацію зображення в більш компактне представлення.

Декодер відповідальний за відновлення зображення зі стиснутого представлення, отриманого енкодером, шлях декодера складається з серії деконволюційних шарів, які поступово збільшують роздільну здатність зображення та зменшують кількість каналів ознак. Цей процес ефективно розгортає стиснуту інформацію, відновлюючи деталі та чіткість зображення.

Архітектура U-Net також включає з'єднання "skip connections", які з'єднують відповідні шари енкодера та декодера. Ці з'єднання дозволяють декодеру отримувати доступ до ознак високої роздільності, видобутих енкодером, забезпечуючи збереження деталей, що визначають його початкову форму [24]. Зображення структури моделі U-Net наведено на рис. 2.3.

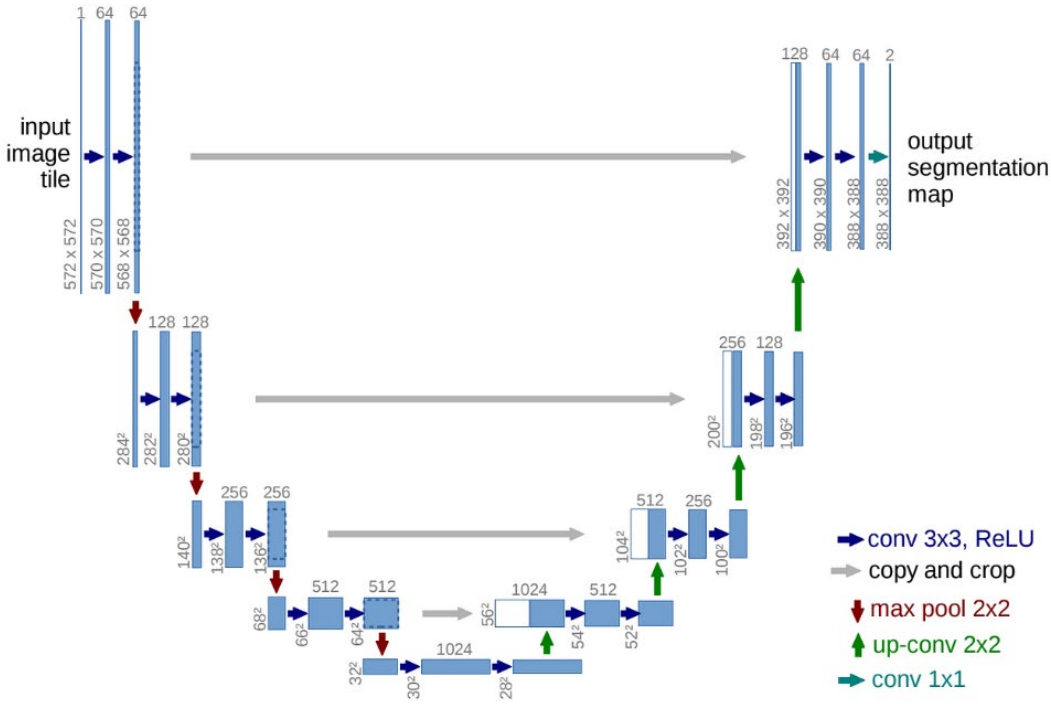

Рис. 2.3. Структура мережі U-Net

Здатність U-Net ефективно вивчати та узагальнювати зображення робить його потужним інструментом для генерації зображень. Навчання U-Net на великому наборі даних парних зображень встановлює зв'язок між представленнями низької та високої роздільності. Це дозволяє U-Net перетворювати зображення низької роздільності, такі як ті, що генеруються шумом або малюнками, в зображення високої роздільності, реалістичні.

Майстерність U-Net у генерації зображень може бути пояснена кількома факторами: ефективний видобуток ознак та точність відновлення зображення. U-Net знайшов широке застосування в різних завданнях генерації зображень у сфері генеративного мистецтва, зокрема:

– підвищення роздільності зображення: U-Net може масштабувати зображення низької роздільності до вищих роздільностей, підвищуючи їх якість та зберігаючи деталі;

– видалення шуму зображення: U-Net може видаляти шум з зображень, відновлюючи їх чіткість та різкість;

– генерація зображень за текстом: U-Net може генерувати зображення на основі текстових описів, перетворюючи мову в візуальні представлення;

– сегментація зображень: U-Net може розділяти зображення на визначені області, ідентифікувати та відокремлювати об'єкти чи області інтересу.

U-Net революціонізував спосіб створення та маніпулювання зображеннями, ставши невід'ємним інструментом у галузі генерації зображень. Його здатність перетворювати представлення низької роздільності в високоякісні зображення відкрила світ можливостей, розширюючи межі творчості та виразності в цифровому просторі [25].

2.3.3. Модель CLIP Text Encoder

CLIP є трансформаторною нейронною мережею, яка навчена на величезному наборі даних тексту та зображень. Набір даних містить пари тексту та зображень, які представляють однакові об'єкти або концепти. Модель CLIP треновано знаходити спільні характеристики між текстом та зображеннями в парі та перетворювати їх у латентні представлення, які мають бути подібні між собою, формуючи пари числових представлень де представлення зображення та представлення його текстового опису значною мірою збігаються. Приклад алгоритму навчання модель CLIP наведено на рис. 2.4.

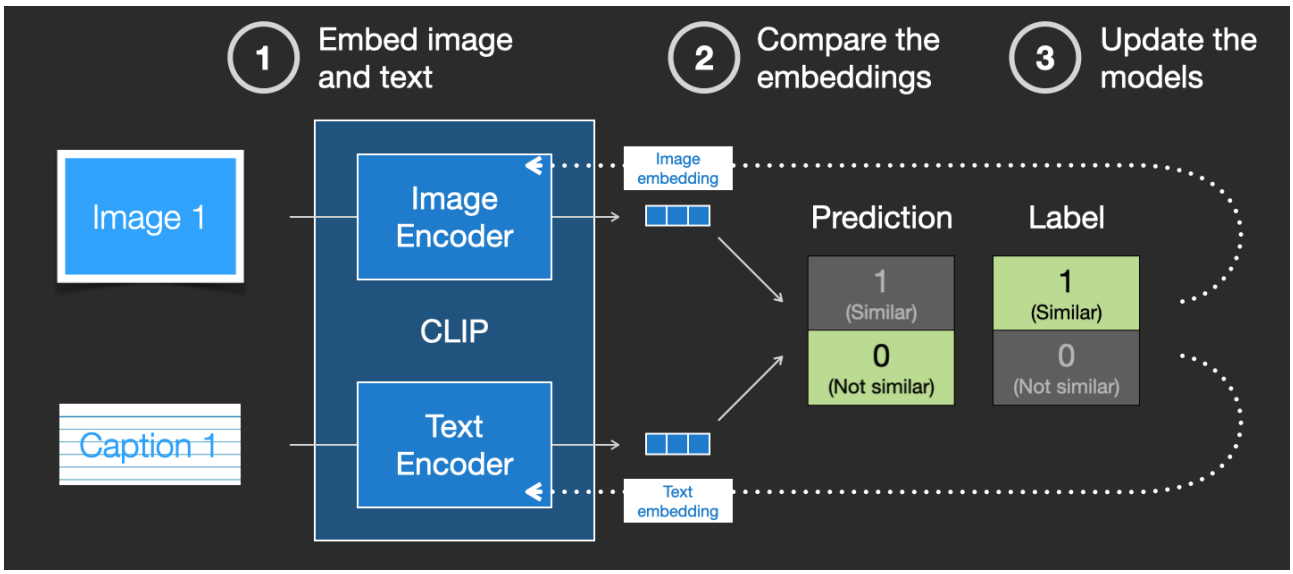

Рис. 2.4. Алгоритм навчання моделі CLIP

CLIP text encoder - це частина моделі CLIP (Contrastive Language-Image Pre-training), яка відповідає за кодування тексту в багатомодальне просторове представлення. Це представлення використовується для навчання моделі CLIP розпізнавати образи та текст, а також для застосування моделі до нових завдань.

Роль CLIP text encoder у моделях латентної дифузії полягає у забезпеченні зв'язку між текстовим описом та зображеннями. Це дозволяє використовувати мову для керування процесом машинного навчання та створення зображень, які відповідають наданому користувачем текстовому опису [26].

Ось кілька конкретних прикладів того, як CLIP text encoder може використовуватися в моделях латентної дифузії:

– кодування текстового опису в векторний простір: CLIP text encoder перетворює текстовий стимул у векторний простір, який є спільним для тексту та зображень. Цей векторний простір дозволяє моделям латентної дифузії розуміти текстовий стимул та використовувати його для керування процесом генерації зображення;

– посилення зв'язку між текстом та зображеннями: CLIP text encoder допомагає моделям латентної дифузії посилити зв'язок між текстом та зображеннями. Це відбувається тому, що CLIP навчається на наборі даних, який

містить парні зображення та їх текстові описи. Це навчання допомагає CLIP навчитися знаходити відповідні зображення для певних текстових описів [27].

CLIP text encoder є потужним інструментом, який може використовуватися у складі моделей латентної дифузії для їх ефективного навчання та використання.

2.4. Навчання моделей латентної дифузії

Латентні дифузійні моделі (LDM) є новим класом генеративних моделей, які за останні роки викликали значний інтерес у галузі штучного інтелекту. Вони здатні генерувати реалістичні та творчі зображення, а також виконувати інші завдання, такі як редагування зображень та створення нових об'єктів.

Процес навчання LDM є складним і багаторівневим. Він передбачає три основні етапи:

– навчання моделі прогнозувати та видаляти шум (denoising);

– навчання моделі генерувати зображення відповідно до текстового опису (text conditioning);

– тонке налаштування (fine-tuning).

Процес тренування починається з підготовки тренувального набору даних. У якості похідного джерела є «чисте» представлення зображення із тренувального набору, згенероване VAE. Окремо створюється шаблон випадкового Гаусівського шуму, який потім розкладається на проміжні кроки від представлення у якому шум відсутній до представлення, яке містить тільки шум. Далі формується новий тренувальний набір – серія зашумлених представлень зображень: до кожного «чистого» зображення додається частка згенерованого раніше шуму (залежно від кількості кроків на які було «розкладено» шум). Таким чином створюється новий набір, який складається з латентних представлень зображень із різним ступенем зашумленості: від чистого зображення до повністю зашумленого [28]. Приклад формування нового тренувального набору із додаванням шуму наведено на рис. 2.5.

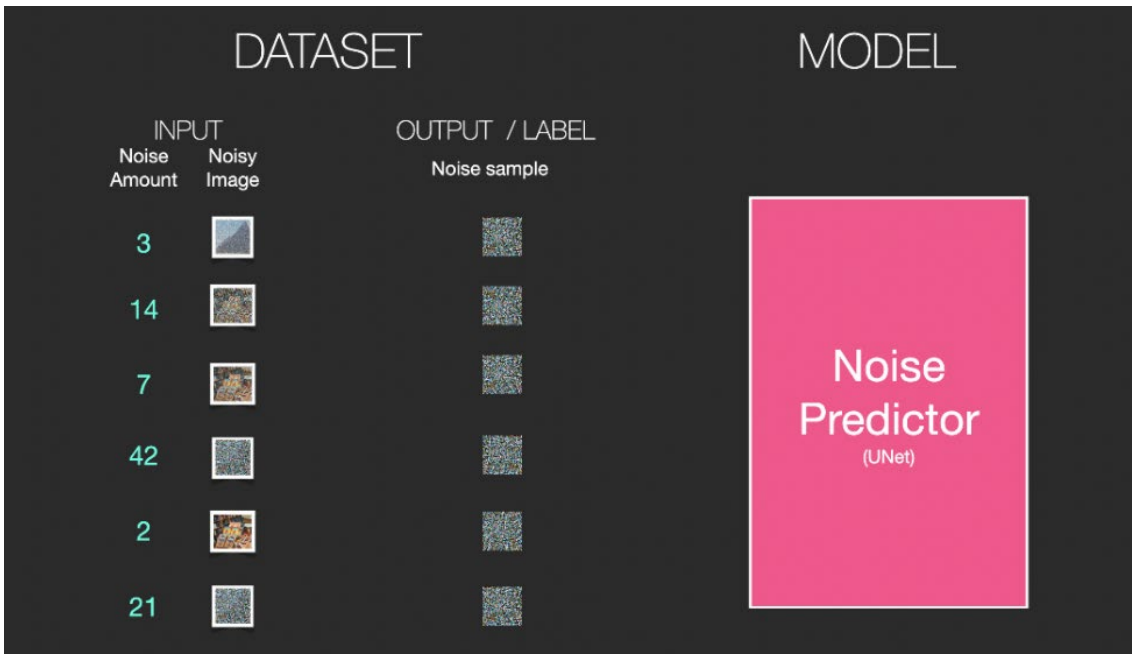

Рис. 2.5. Формування нового тренувального набору із додаванням шуму

Після того як було створено тренувальний набір, можна переходити до навчання моделі U-Net прогнозувати та видаляти шум із зображень. Модель вчиться оцінювати та прогнозувати наступний крок у процесі видалення шуму, починаючи з дуже зашумленого зображення і поступово відновлюючи його до його початкового стану. Спостерігаючи за відмінностями між цими парами, модель навчається прогнозувати наступний крок у процесі видалення шуму. Це передбачає ітеративне видалення шуму з шумних зображень, поступово відновлюючи їх до їхнього початкового стану.

Цей процес передбачає ітеративне застосування функції видалення шуму до зображення. Функція видалення шуму може бути будь-якою, але часто використовується метод середнього квадратичного відхилення:

1. Модель прогнозує наступний крок у процесі видалення шуму;
2. Модель обчислює різницю між прогнозом та фактичним зображенням із тренувального набору (target);
3. Модель оновлює свої параметри, щоб зменшити цю різницю.

Для керівництва цим процесом видалення шуму використовується функція втрат. Ця функція вимірює розбіжність між прогнозами моделі та контрольними чистими зображеннями. Параметри моделі потім коригуються для

мінімізації цієї втрати, гарантуючи, що її можливості видалення шуму продовжують покращуватися. Приклад тренування моделі U-Net для прогнозування доданого шуму наведено на рис. 2.6.

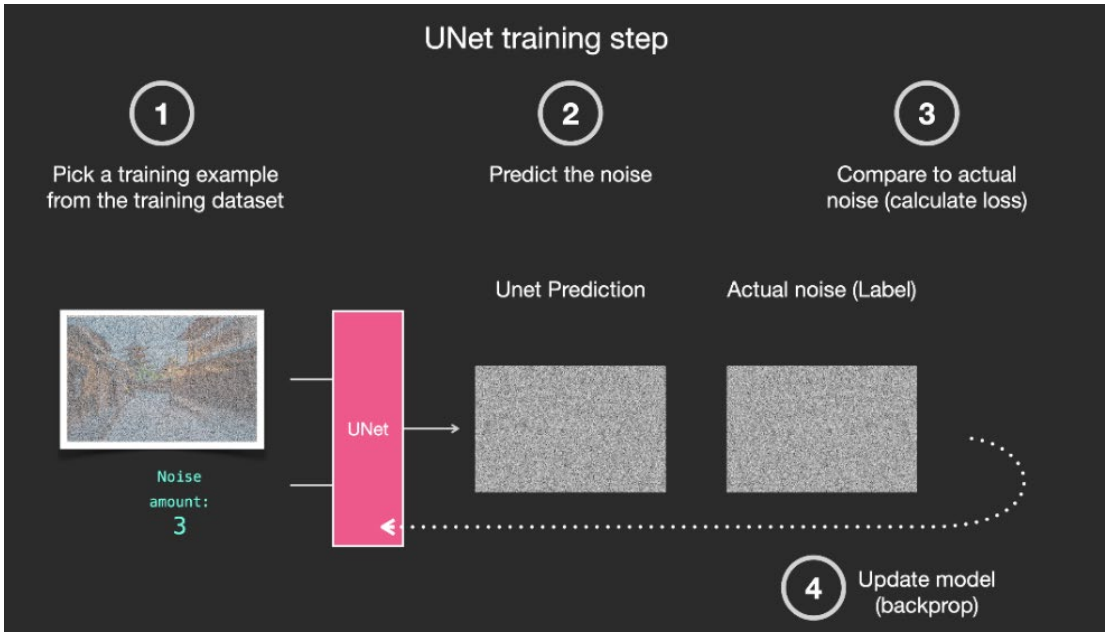

Рис. 2.6. Тренування моделі U-Net для прогнозування доданого шуму

Підсумовуючи, можна сказати, що на цьому кроці модель навчається базовим основам генеративного навчання. Вона вчиться розуміти, як шум впливає на зображення, і як видалити цей шум, не пошкодивши зображення. Цей процес є важливим фундаментом для подальшого навчання моделі [29].

Фактори, які впливають на якість навчання моделі:

– використання високоякісних даних: чим якісніші будуть дані, тим краще модель навчиться видаляти шум;

– використання відповідної функції втрат. Функція втрат повинна бути достатньо чутливою, щоб модель могла навчитися видаляти шум, але не настільки чутливою, щоб вона почала генерувати артефакти;

– виконувати достатню кількість ітерацій, адже для ефективного навчання видаленню шуму може знадобитися багато ітерацій.

На другому етапі навчання модель навчається генерувати зображення на основі текстових описів. Цей процес називається текстовим кондиціонуванням.

Для навчання текстового кондиціонування модель тренується на наборі зображень та їх текстових описів. Модель навчається прогнозувати наступний крок у процесі генерації зображення, починаючи з текстового опису і поступово створюючи зображення, яке відповідає цьому опису. Цей процес аналогічний процесу навчання видаленню шуму, але тут модель також використовує текстовий опис для генерації зображення.

Автоенкодер створює латентне представлення вхідного тренувального зображення, модель CLIP Text encoder перетворює наведений опис у його латентне представлення – числовий вектор, який тепер може бути інтегровано у робочий латентний простір моделі. Коли обидва представлення (текст та зображення) сформовані, їх можна об’єднати у спільне латентне представлення за допомогою механізму cross-attention, який дозволяє моделі ефективно отримувати інформацію із двох різних джерел (рис. 2.7.).

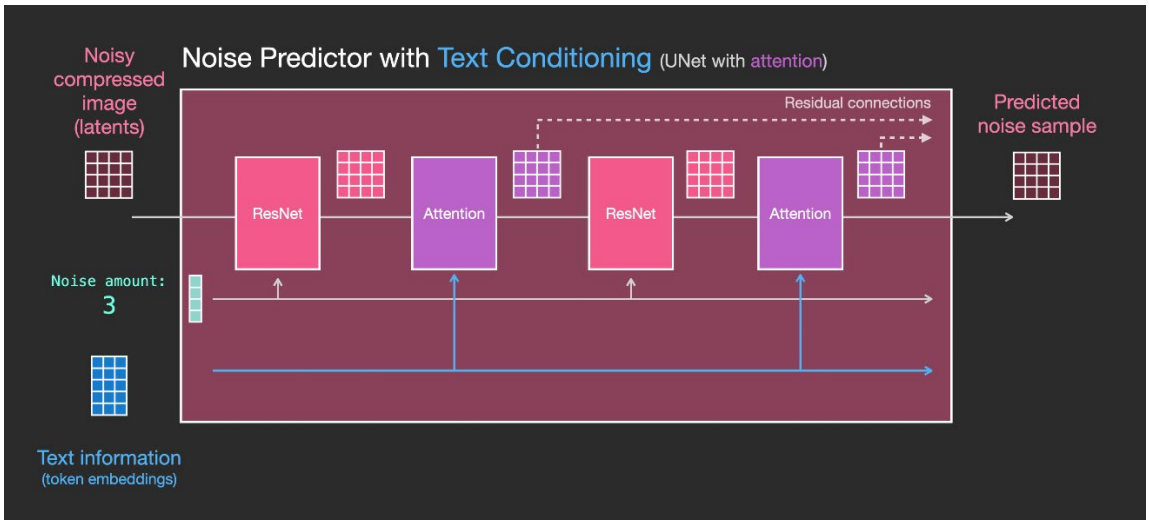

Рис. 2.7. Навчання моделі прогнозувати шум із вказівкам із текстового опису

Під час навчання модель вчиться асоціювати представлення вхідного зображення із представленням його текстового опису. Цей процес повторюється багато разів, поки модель не навчиться генерувати зображення, які відповідають текстовим описам із високою точністю.

Останнім етапом навчання є тонке налаштування. На даному етапі навчання модель використовується для генерування зображень із заданими стилями або творчими результатами.

Для тонкого налаштування модель тренується на наборі даних зображень, які мають бажані стилі або творчі результати (наприклад, зображення конкретних об’єктів). Модель навчається прогнозувати наступний крок у процесі генерації зображення, починаючи з випадкового зображення і поступово перетворюючи його в зображення із заданим стилем або творчим результатом.

Цей процес аналогічний процесу навчання видаленню шуму та текстового кондиціонування, але тут модель також використовує інформацію про бажаний стиль або творчий результат.

2.5. Генерація зображень із текстового опису

Метод використання генеративних моделей, заснованих на дифузії - це алгоритм генеративного навчання, який працює шляхом формування початкового випадкового шуму та його подальшого поступового очищення до тих пір, поки він не перетвориться на дані, які схожі на ті, що були в тренувальному наборі даних, на якому навчено модель. Розглянемо алгоритм генерування зображення на основі текстового опису (text-to-image) із використанням моделі латентної дифузії.

Першим кроком у створенні зображення на основі текстового опису є створення тензора у прихованому просторі (latent space). Цей тензор містить шум, який буде використано для створення зображення, він уявляє собою Гаусівський шум (випадковий шум, який має нормальний розподіл) [30]. Приклад візуалізації початкового шуму виглядає наступним чином наведено на рис. 2.8.

Рис. 2.8. Візуалізація початкового шуму

На наступному кроці Text Encoder мовної моделі CLIP перетворює вхідний текстовий опис (Prompt) у латентний вектор, який представляє семантичне значення кожного слова/токену у запиті (кожному токену відповідає окремий вектор). Даний вектор буде об’єднано із представленням шуму та інтегровано у латентний простір для керування моделлю під час генерації, щоб забезпечити відповідність генерованого зображення текстовому опису. Цим займається cross-attention механізм, який дозволяє узгодити представлення з двох різних джерел [31].

Під час генерації для відповідності вихідного зображення текстовому опису, модель на кожному кроці створює два окремих зображення із поточного латентного представлення: одне, яке кероване вектором-представленням текстового опису і друге – порожнє. Потім модель оцінює яке з отриманих представлень більше відповідає бажаному зображенню. Модель продовжує роботу із представленням, яке було оцінено як більш відповідне.

Тепер, отримане об’єднане латентне представлення використовується у якості вхідних даних для нейронної моделі U-Net, яка «прогнозує» або «оцінює» шум у вхідному наборі та поступово видаляє його. Тобто, входом для даної мережі є зашумлений латентний простір, а виходом – шум прогнозований у даному латентному просторі. Візуалізацію кроків очищення представлення генерованого зображення від шуму наведено на рис. 2.9.

Рис. 2.9. Візуалізація кроків очищення зображення від шуму

Метод, що використовується під час видалення шуму називається методом вибірки (sampling method), оскільки на кожному кроці виконується створення нового зображення-зразка (sample). Є множина доступних методів вибірки, кожен з яких у тій чи іншій мірі впливає на процес видалення шуму та фінальне зображення. Кожен метод описується математичною формулою, яка визначає як саме буде очищено шум (частка від загального обсягу, скільки попередніх кроків обробки буде враховано, чи слід додавати новий випадковий шум із метою підвищення варіативності вихідного зображення).

Після того як оцінений шум було видалено з зображення, виконується декодування латентного простору та повернення у піксельний простір (перетворення у піксельне зображення більшої роздільної здатності) [32]. Спрощену схема генерації зображення наведено на рис. 2.10.

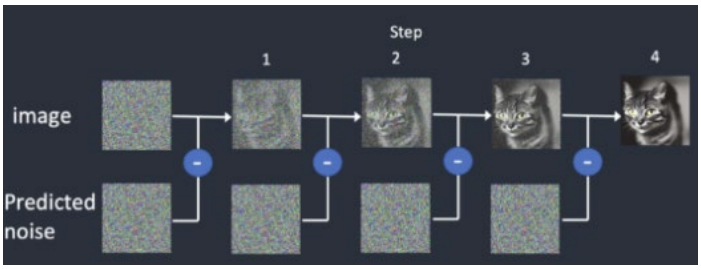

Рис. 2.10. Спрощена схема генерації

Таким чином, використання моделей латентної дифузії є одним із найпоширеніших методів генеративного навчання, який використовується для створення реалістичних зображень. Він працює шляхом поступового розм'якшення випадкового шуму до кінцевого зображення.

2.6. Основні параметри генерації

Для початку процесу генерування зображення користувачу потрібно виконати певні налаштування. Частка цих налаштувань є явною та очевидно визначає зображення, що буде отримано у результаті, проте є й такі параметри, які не є явними, та їх підбір та корегування можуть бути виконані емпірично на основі вже отриманих результатів. Розглянемо кожний із базових параметрів окремо та визначимо його властивості та призначення у процесі генерації зображення:

1. Model/Checkpoint. У основі генерації зображень знаходяться попередньо навчені моделі, які також часто називають файлами контрольної точки (Checkpoint). Ці моделі складаються з вагів, попередньо навчених для використання методом стабільної дифузії, призначених для створення або загальних візуальних образів, або зображень в певному жанрі.

Вибір моделі ключовим чином впливатиме на отримані результати, тому її підбір є одним із ключових етапів. Існує безліч навчених моделей, які було натреновано для генерування певних зображень: від фотореалістичних портретів та пейзажів до абстрактних ілюстрацій. Можна знайти вже навчені моделі на відповідних ресурсах, або натренувати модель самому, що потребує значної обчислювальної потужності та підготовку тренувального набору даних. Тому якщо перед вами не стоїть занадто специфічна задача, то буде зручніше просто знайти підходящу навчену модель, яку було треновано для генерації необхідних вам об’єктів, стилів, тощо [33].

Типи зображень, які модель може генерувати, визначаються даними, що використовуються під час її процесу навчання. Наприклад, якщо модель ніколи не стикалася з котом під час навчання, вона не зможе створити зображення кота. Точно так само, якщо дані для навчання складаються виключно з зображень котів, модель буде виключно генерувати візуальні образи котів [34].

– 2. Prompt – це запит до моделі, тобто текстовий опис зображення, яке потрібно створити. Він використовується для надання моделі додаткової

інформації про те, що вона повинна створити. Prompt використовується для поліпшення якості генерованих зображень. Він допомагає моделі зрозуміти, що саме потрібно створити [35].

– 3. Negative Prompt - це текстовий опис зображення або його елементів, які не потрібно створювати. Він використовується для обмеження можливостей моделі. Negative Prompt використовується для запобігання створенню небажаних елементі та деталей зображень. Він допомагає моделі зосередитися на створенні бажаних користувачем зображень.

4. Sampling method – це метод, який буде використовуватися для видалення шуму під час генерації зображення з використанням Stable Diffusion checkpoint і prompt. Для створення зображення "Stable Diffusion" спочатку генерує абсолютно випадкове зображення в латентному просторі. Потім предиктор шуму оцінює шум зображення та «передбачений» шум віднімається від зображення. Цей процес повторюється декілька разів, та у результати ви отримуєте чисте зображення [36].

Цей процес очищення від шуму називається вибіркою, оскільки "Stable Diffusion" генерує нове зображення на кожному кроці. Метод, використовуваний у вибірці, називається «семплером» або методом вибірки.

Є множина доступних Sampling-методів, їх вибір також буде напряму впливати на генероване зображення.

Різні методи очищення можуть мати різну специфіку використання, проте здебільшого методи, що наслідують один і той самий алгоритм дають схожі результати. Розглянемо базові алгоритми очищення, які появилися одними із першими та були використані у складі моделей латентної дифузії:

– Euler - Найпростіший можливий метод – метод Ейлера. Це базовий метод для чисельного інтегрування диференціальних рівнянь.

– Heun - Більш точна, але повільна версія методу Ейлера. Цей метод використовує корекційні кроки для поліпшення точності результатів порівняно з методом Ейлера.

– LMS (Лінійний багатокроковий метод) - має ту ж швидкість, що і метод Euler, але, орієнтовно, забезпечує більшу точність. Цей метод використовує інформацію з кількох попередніх кроків для оновлення поточного значення, що може призвести до поліпшення точності порівняно з однокроковими методами, такими як Euler.

Також, розглядаючи перелік доступних методів, можна побачити, що їх назви частково збігаються, проте мають трохи різне найменування, наприклад методи з літерою «a» на кінці – так звані предківські/наслідувані (ancestral) методи, такі як, наприклад, Euler a, DPM2 a, DPM++ 2S a або DPM++ 2S a Karras.

Це, такі звані, предківські семплери. Предківський семплер додає шум до зображення на кожному кроці вибірки. Вони є стохастичними семплерами, оскільки результат вибірки має певну випадковість. Слід мати на увазі, що багато інших також є стохастичними семплерами, навіть якщо їхні назви не містять літеру "a".

Недолік використання предківського семплера полягає в тому, що зображення не буде збігатися. Порівняння процесу генерування зображень за допомогою методів Euler та Euler a, наведено на рис. 2.12.

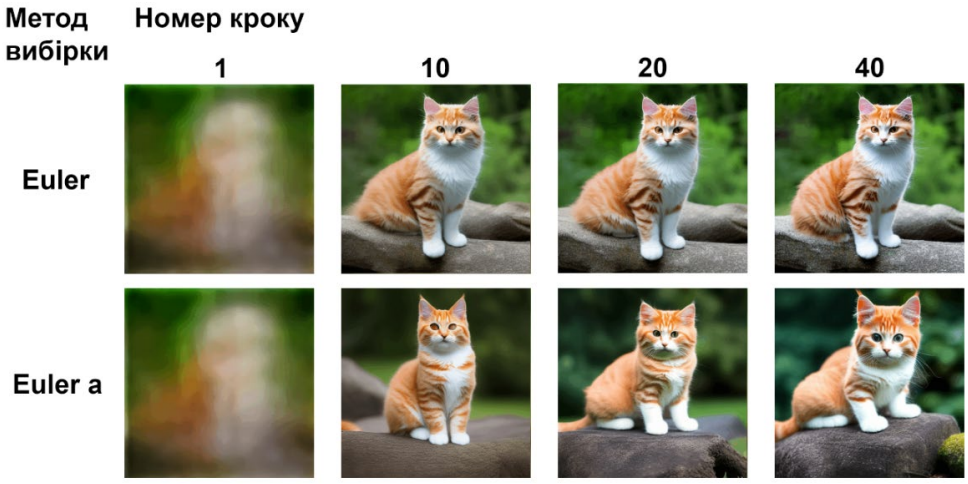

Рис. 2.12. Порівняння зображень згенерованих методами Euler та Euler a.

Зображення, згенеровані за допомогою методу Euler a, не збігаються при великій кількості кроків вибірки. Натомість зображення з методу Euler добре збігаються (рис. 2.13.). Для відтворюваності бажано, щоб зображення збігалися.

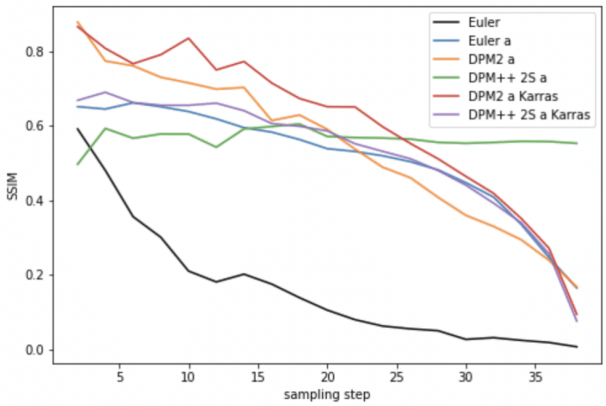

Рис. 2.13. Порівняння збігання зображень із використанням предківських методів вибірки

Через використання алгоритму додавання випадкового шуму, деякі методи можуть давати різні вихідні результати на великій кількості кроків обробки (адже вплив ймовірностей зростає). Приклад збігання зображень при повторній генерації кожним з методів Euler, DDIM, PLMS, Heun та LMS Karras наведено на рис. 2.14.

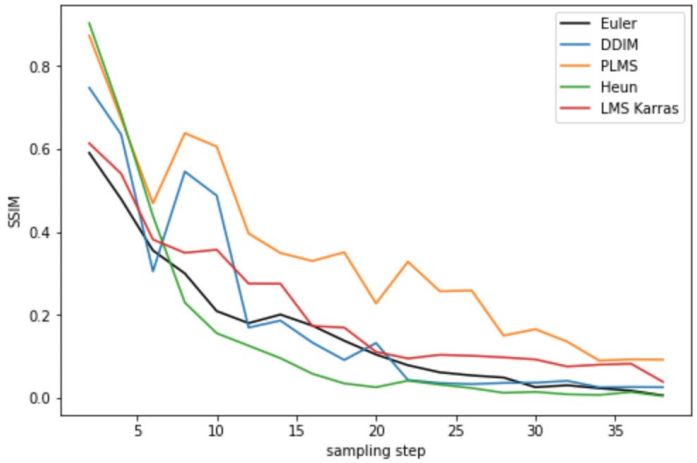

Рис. 2.14. Збіжність зображень при повторній генерації кожним з методів Euler, DDIM, PLMS, Heun, LMS Karras

Приклад того як використання різних алгоритмів очищення шуму впливає на кінцеве зображення (за умови що усі інші налаштування залишаються незмінними) наведено на рис. 2.15.

Рис. 2.15. Вплив обраного методу очищення шуму на генероване зображення

Далі трохи детальніше про основні методи:

– Euler a - семплер схожий на семплер Euler. Але на кожному кроці він віднімає більше шуму, ніж має, і додає деякий випадковий шум для відповідності планувальнику шуму (noise scheduler). Очищене від шуму зображення залежить від конкретного шуму, доданого на попередніх кроках. Таким чином, це предківський семплер в тому сенсі, що шлях очищення від шуму зображення залежить від конкретних випадкових шумів, доданих на кожному кроці. Результат буде відмінним, якщо ви вирішите виконати цей процес ще раз.

– Heun - метод Гойна є більш точним вдосконаленням методу Ейлера. Але потрібно виконувати прогнозування шуму двічі за кожен крок, тому він вдвічі повільніший за Ейлера;

– DDIM (Denoising Diffusion Implicit Models) та PLMS (Pseudo Linear Multi-Step method) були одними з перших розроблених методів вибірки, наразі не рекомендуються до використання;

– LMS семплери - Подібно до методу Ейлера є стандартним методом для вирішення звичайних диференціальних рівнянь. Він спрямований на підвищення точності за допомогою розумного використання значень попередніх кроків часу;

– DPM та DPM++ семплери - розв'язники дифузійних ймовірнісних моделей (DPM-Solvers), які належать до сімейства новорозроблених методів очищення для моделей дифузії. Наразі є найбільш вживаними методами вибірки.

Також слід мати на увазі, що усі методи тим чи іншим чином унікальні у своїх роботі, а отже потребують різної кількості часу на генерацію. Наприклад, алгоритми очищення групи DPM потребують більше часу на виконання кожної ітерації, порівняно із іншими. Кількість часу, витрачену на генерацію набору з 8 картинок різними алгоритмами очищення шуму наведено на рис. 2.16.

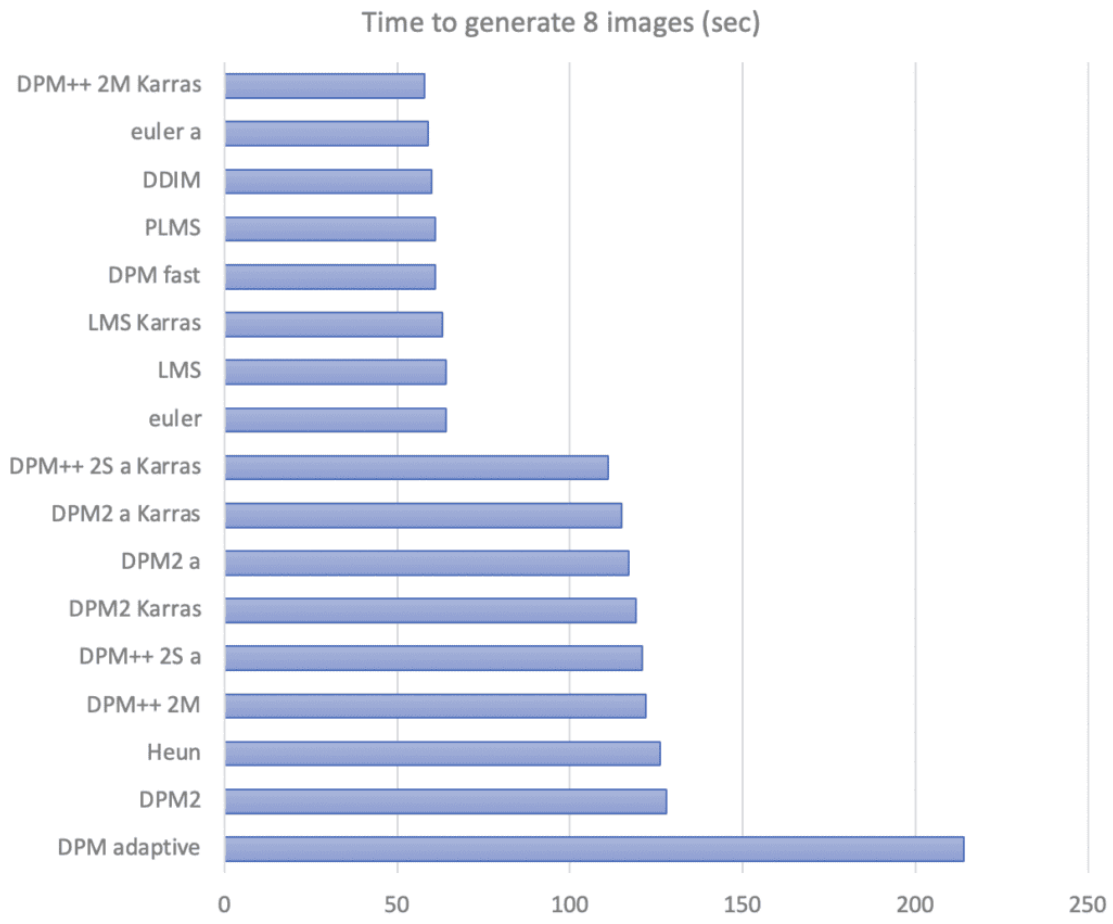

Рис. 2.16. Час у секундах, витрачений на генерування набору з 8 зображень при використанні різних алгоритмів очищення шуму

4. Sampling steps – кількість кроків обробки - визначає кількість ітерацій генерації зображення після створення шуму. Якість зростає при збільшенні кількості кроків вибірки. Зазвичай 20 кроків із семплером Ейлера вже достатньо

для отримання високоякісного, чіткого зображення. Хоча зображення буде незначно змінюватися при переході до вищих значень, воно може стати іншим, але не обов'язково вищої якості. Зазвичай рекомендовано використовувати даний параметр у діапазоні 20-30. Якщо якість вихідного зображення вас не влаштовує, то можна збільшити кількість кроків обробки. Приклад впливу збільшення кількості кроків обробки на вихідні зображення наведено на рис. 2.17.

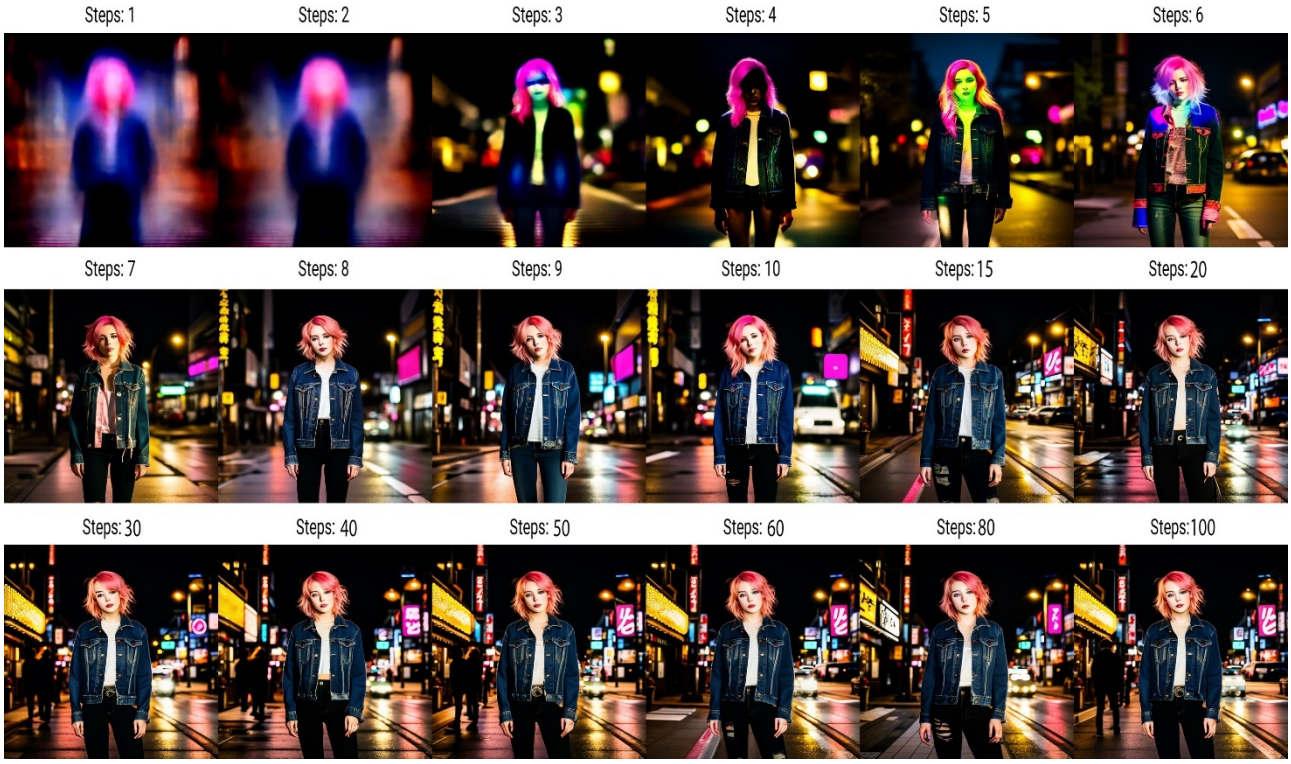

Рис. 2.17. Вплив збільшення кількості кроків обробки на вихідні зображення

Але не слід очікувати, що якість генерованих зображень буде нескінченно зростати відповідно до кількості кроків. Насправді, у більшості випадків при занадто високих значеннях кількості кроків обробки (після 40) на зображенні можуть почати з’являтися так звані артефакти, до того ж структура та композиція зображення можуть не відповідати очікуваним або отриманим на меншій кількості кроків, тому не слід виходити за рекомендовані межі, адже це не дає значної прибавки у якості, проте потребує більшої обчислювальної потужності.

5. Width and height - розмір вихідного зображення. Оскільки більшість моделей Stable Diffusion навчаються на зображеннях розміром 512×512,

відхилення від цього розміру може викликати проблеми, такі як дублювання об'єктів. Залиште його квадратним, якщо це можливо. Для портретних зображень рекомендовано 512×768 або 768×512 для ландшафтних. Також слід мати на увазі, шо навіть мінімальна зміна розміру зображення призведе до того, що ви отримаєте нове зображення, а не просто збільшену версію зображення оригінального розміру. Для збільшення ориганільного зображення є спеціальні інструменти, які буде розглянуто пізніше.

6. CFG scale: Classifier Free Guidance scale є критичним параметром у Stable Diffusion, який контролює те, наскільки згенероване зображення відповідає вхідному запиту користувача. Простіше кажучи, він визначає, наскільки ШІ «слухає» підказку.

Процес генерації зображення з тексту Stable Diffusion полягає в послідовному уточненні шумного зображення до тих пір, поки воно не відповідатиме текстовому опису із запита користувача. Параметр CFG впливає на вагу цієї відповідності. Вище значення CFG призводить до того, що модель більш точно слідує підказці, тоді як нижче значення дозволяє більше творчої свободи та сприяє утворенню більш креативних зображень.

Параметр CFG зазвичай налаштовується між 1 і 30, де 1 – максимальна творча свобода моделі, а 30 – строге слідування введеному запиту. Значення 7 часто вважається оптимальною стартовою точкою, тому навіть розробники залишили його у якості стандартного значення, оскільки він забезпечує баланс між точністю відповідності підказці та творчим пошуком.

Приклади використання різних значень параметра CFG:

– значення CFG = 1: модель має мінімальне керівництво і створює дуже творчі, але часто абстрактні зображення, які можуть не відповідати підказці;

– значення CFG = 7: модель знаходить баланс між дотриманням підказки та вивченням творчих варіацій. Це часто вважається оптимальною стартовою точкою, яка дозволяє дослідити вплив текстового опису на вихідне зображення;

– значення CFG = 15: модель більш точно дотримується підказки, створюючи зображення, які часто більш реалістичні, але менш творчі.

Може здатися, що чим більше значення, тим краще, адже очікується що модель буде ідеально дотримуватися запиту, однак на практиці це не зовсім так. При занадто високих значення CFG на зображенні з’являються артефакти, руйнується його структура, композиція та кольорова палітра. Приклад того як змінюється вихідне зображення відповідно до підвищення значення параметра CFG наведено на рис. 2.18.

Рис. 2.18. Зміна вихідних зображень відповідно до збільшення значення параметра CFG

Параметр CFG є лише одним із багатьох факторів, які впливають на генерацію зображень. Інші параметри, такі як кількість кроків та графік шуму, також відіграють значну роль. Параметр CFG - це потужний інструмент, який можна використовувати для контролю творчого стилю та точності зображень, згенерованих Stable Diffusion. Розуміючи, як працює цей параметр і експериментуючи з різними значеннями, ви можете досягти широкого спектра

результатів - від дуже творчих та абстрактних зображень до реалістичних та детальних зображень вашої уяви.

7. Seed - це числове значення, що використовується для ініціалізації процесу генерування зображень. Воно служить відправною точкою для моделі, визначаючи початковий випадковий шум, з якого генерується зображення. Задаючи певне значення seed, ви можете контролювати відтворюваність згенерованих зображень, гарантуючи, що ви зможете відтворити те саме зображення з тим самим текстовим запитом та іншими налаштуваннями.

Значення seed зазвичай представлено як 32-бітове ціле число. Ви можете вказати конкретне значення seed в інтерфейсі Stable Diffusion або залишити його порожнім, щоб генерувати випадкове. Якщо ви хочете відтворити певне зображення, ви можете скопіювати значення seed з інформації про згенероване зображення.

Під час генерації зображення за допомогою Stable Diffusion, модель починає з шаблону випадкового шуму. Потім цей шаблон шуму поступово очищається через ряд кроків, керованих текстовим запитом та різними параметрами. Seed визначає початковий стан цього шаблону шуму, впливаючи на загальний вигляд та характеристики згенерованого зображення [36].

Параметр seed відіграє вирішальну роль у Stable Diffusion з кількох причин:

– відтворюваність: використання одного й того ж значення seed завжди генеруватиме одне й те саме зображення, за умови використання того самого текстового запиту та налаштувань. Це корисно для експериментів з різними параметрами та точного налаштування процесу генерування зображень.

– творче дослідження: експериментуючи з різними значенням параметра seed, ви можете дослідити широкий спектр варіацій того самого текстового запиту, що призводить до несподіваних і творчих результатів.

– спільний доступ та співпраця: надання значення seed разом із текстовим запитом дозволяє іншим відтворити ваше зображення та спільно працювати над подальшими ітераціями.

Використання різних значень seed призведе до того, що згенеровані зображення будуть різними. Однак, слід зазначати, що використовуючи одне й те саме значення seed, при цьому незначним чином модифікуючи запит, можна у більшості випадків очікувати, що нове отримане зображення буде містити схожу до оригінального зображення композицію (схоже розташування об’єктів, їх положення тощо). Приклад модифікації текстового опису при ідентичному значенні seed наведено на рис. 2.19.

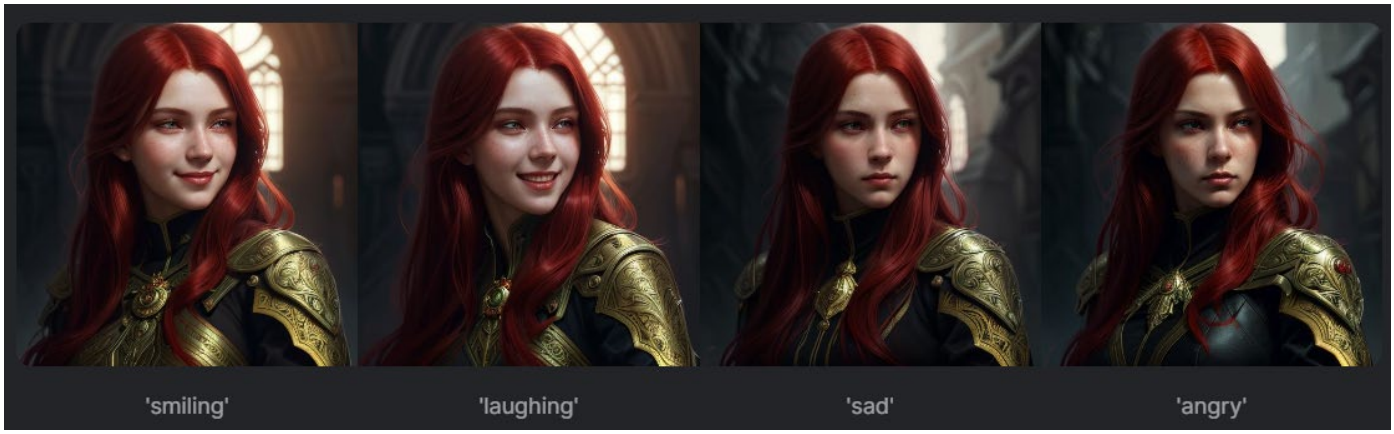

Рис. 2.19. Генерування зображень із модифікованими текстовими описами, але ідентичним значенням seed

Можна помітити, що зображення схожі між собою, однак відповідно до запиту було змінено вираз обличчя/емоцію на зображеннях. Використання одного й того ж значення seed дозволяє значною мірою відтворити оригінальне зображення, при цьому вносячи до нього бажані змін.

2.7. Особливості моделей латентної дифузії

Використання моделей латентної дифузії має ряд особливостей, які роблять їх ефективними для використання у сфері генеративного мистецтва:

– створення зображень високої якості. Моделі латентної дифузії можуть генерувати зображення, які часто неможливо відрізнити від зображень, створених людьми;

– відносна простота використання. Моделі латентної дифузії є відносно простими у використанні, що робить їх доступними для широкого кола користувачів;

– можливість генерувати різні типи зображень. Моделі латентної дифузії можна використовувати для генерації різних типів зображень, включаючи реалістичні зображення, абстрактні зображення та ілюстрації.

Однак, використання моделей латентної дифузії також має ряд недоліків:

– можуть створювати зображення, які є занадто схожими між собою. Це пов'язано з тим, що алгоритм навчається на наборі даних зображень, які часто мають схожі риси;

– можуть бути повільним. Моделі латентної дифузії може бути повільним, оскільки під час процесу генерації модель повинна поступово розмивати шум, щоб створити зображення;

– потребують певної обчислювальної потужності від машини, що накладає певні обмеження на апаратну складову [36].

2.8. Обмеження та вимоги використання моделей латентної дифузії

Хоча моделі латентної дифузії за рахунок своєї інноваційної архітектури і мають значні переваги порівняно із класичними дифузійними моделями, швидкість та можливості роботи моделей латентної дифузії можуть обмежуватися такими характеристиками графічної карти, як:

– GPU Core Clock Speed (Швидкість ядра GPU) - Швидкість ядра GPU визначає, наскільки швидко GPU може обробляти інструкції та виконувати обчислення. Вища швидкість ядра веде до швидших часів обробки, що є важливим для генерації зображень за допомогою Stable Diffusion. Складні алгоритми перетворення текстових описів в високоякісні зображення вимагають швидкої обробки з боку GPU. Вища швидкість ядра гарантує, що GPU може ефективно обробляти ці обчислення, що призводить до швидкої генерації зображень і більш гладкого функціонування.

– Обсяг пам'яті GPU - пам'ять GPU, також відома як GPU RAM, служить тимчасовим сховищем для даних, які GPU повинен обробляти під час генерації зображень. До цих даних входять текстові описи, проміжні представлення зображень та різні параметри, пов'язані з процесом генерації зображень. Обсяг доступної пам'яті GPU безпосередньо впливає на складність та розмір зображень, які можуть бути згенеровані.

З достатньою кількістю пам'яті GPU модель латентної дифузії може впоратися з витонченими обчисленнями та зберігати проміжні дані зображення, необхідні для створення високороздільних і деталізованих зображень. Однак якщо пам'яті GPU недостатньо, можуть виникнути проблеми з пам'яттю, що призводять до повільної обробки, артефактів зображення чи навіть аварій. Таким чином, GPU з достатньою пам'яттю є важливим для досягнення оптимальної продуктивності та генерації зображень високої якості.

– обсяг VRAM (відеопам'яті) - VRAM, спеціально розроблений для обробки графіки, є підмножиною пам'яті GPU, яка відіграє важливу роль у генерації зображень. Він зберігає текстури, інформацію про кольори та інші візуальні дані, які безпосередньо відображаються на екрані. Stable Diffusion широко використовує VRAM для відображення проміжних і кінцевих представлень зображень. Достатня кількість VRAM забезпечує, що GPU може впоратися з витонченими текстурами та даними про кольори, пов'язаними з генерацією високороздільних зображень. Якщо VRAM обмежено, система може використовувати повільну пам'ять, що може призвести до зниження продуктивності та потенційного погіршення якості зображення. Таким чином, GPU із достатньою кількістю VRAM є важливим для генерації зображень у реальному часі та забезпечення високоякісного виводу зображень;

– GPU Memory Bandwidth (Пропускна здатність пам'яті GPU) - пропускна здатність пам'яті GPU вимірює швидкість, з якою дані можуть бути передані між пам'яттю GPU та її обчислювальними ядрами. Цей обмін даними відіграє важливу роль у генерації зображень, оскільки він включає переміщення великих обсягів даних зображення, текстових описів та проміжних представлень

вздовж обчислювального конвеєра. Достатня пропускна здатність пам'яті забезпечує, що GPU має необхідний пропуск для обробки неперервного руху даних, що призводить до швидшої генерації зображень та більш гладкого функціонування. Якщо пропускна здатність пам'яті недостатня, GPU може стикнутися з проблемами, спричиняючи затримки в обробці та, можливо, викликаючи артефакти зображення чи дефекти відтворення;

– одиниці обчислення GPU (CUs) - це фундаментальні обчислювальні блоки в графічному процесорі NVIDIA. Кожна CU містить численні потокові мультипроцесори (SM), які виконують інструкції, необхідні для генерації зображення. Вища кількість CUs, як правило, вказує на більшу потужність обчислень та можливості паралельної обробки.

У контексті використання моделі латентної дифузії більша кількість CUs дозволяє GPU розподіляти складні обчислення, пов'язані з генерацією зображень, на кілька обчислювальних блоків. Цей підхід паралельної обробки значно підвищує швидкість та ефективність процесу генерації зображень, що призводить до швидшого відтворення зображень та більш відзивчивого користувацького досвіду;

– одиниці обробки тензорів GPU (TPUs) - це спеціалізовані прискорювачі штучного інтелекту, призначені для обробки операцій над матрицями, які є важливими для завдань машинного навчання, таких як генерація зображень. TPUs оптимізовані для виконання множення матриць та інших операцій з тензорами, які є основою алгоритмів, використовуваних у Stable Diffusion [37].

Хоча для роботи з моделями латентної дифузії TPUs не є строго обов'язковими, GPU з TPUs можуть забезпечити значне покращення продуктивності, особливо для складних чи високороздільних завдань генерації зображень

2.9. Функції моделей латентної дифузії та розширення, сумісні з ними

Командою розробників оригінальної моделі стабільної дифузії Stable Diffusion, а також розробниками-ентузіастами, які долучилися до даного проекту було створено значну кількість функціональних компонентів, що доступні користувачу, які надають змогу виконувати різноманітні операції на майже будь-якій стадії генерації зображення, починаючи від розширення та додавання методів прибирання шуму, додатків для редагування, аналізу та автоматизації процесу написання запиту до моделі до різноманітних методів та засобів пост-обробки створених або існуючих зображень із метою їх модифікування та покращення. У даному підпункті буде розглянуто ключові компоненти, які дозволяють спростити робочий процес, надати більшу гнучкість процесу генерації та обробці отриманих зображень.

2.9.1. Text-To-Image

Функція Text-to-Image є потужним інструментом, який дозволяє користувачам генерувати реалістичні та креативні зображення за допомогою текстових описів. При генерації зображень у даному режимі, генерація виконується на основі наданого користувачем текстового опису. У даному режимі працює шляхом поступового розмиття випадкового шуму до тих пір, поки він не перетвориться на зображення, яке буде віжповідати наданому текстовому опису, базуючись на зображеннях в наборі даних, на якому було треновану модель латентної дифузії [38].

Основні властивості функції text-to-image у Stable Diffusion:

– реалізм. Модель латентної дифузії може генерувати зображення, які вражають реалістичністю і часто не відрізняються від реальних фотографій;

– креативність: модель може створювати високо креативні та уявні зображення, відображаючи унікальні ідеї та описи користувача;

– контроль: користувачі мають ступінь контролю над процесом генерації зображень, регулюючи параметри та надаючи докладніші текстові запити;

– доступність: функція text-to-image є відносно простою у використанні, що дозволяє генерувати зображення користувачам з різними рівнями навичок та досвіду.

2.9.2. Image-To-Image

Функція image-to-image є вражаючим інструментом, який дозволяє користувачам перетворювати існуючі зображення в нові та творчі варіації. На відміну від функції text-to-image, яка генерує зображення з нуля на основі текстових описів, функція image-to-image модифікує та покращує наявні зображення за допомогою вказівки у вигляді початкового зображення чи додаткових текстових запитів.

Процес генерація зображення у режимі Image-To-Image починається з введення початкового зображення - користувач надає початкове зображення як основу для процесу трансформації зображення. Далі виконується аналіз початкового зображення: початкове зображення аналізується для виділення його особливостей, таких як форми, кольори та текстури. Початкове зображення трансформується через серію модифікацій, під час яких до представлення вхідного зображення алгоритмом додається певна (визначена користувачем) кількість випадкового шуму, після чого модель прогнозує та видаляє шум із представлення, керуючись виділеними особливостями та додатковим текстовим описом. Процес трансформації зображення виробляє кілька варіацій початкового зображення, кожна з унікальними художніми тлумаченнями та творчими елементами [39].

До основних характеристики функції image-to-image при роботі із моделями латентної дифузії можна віднести наступні особливості:

– збереження оригінального зображення: функція image-to-image зберігає основні елементи початкового зображення, додаючи творчі варіації та покращення;

– творче дослідження: функція дозволяє користувачам вивчати різні художні напрямки, перетворюючи зображення в різні стилі, жанри або настрої;

– детальний контроль: користувачі можуть контролювати процес трансформації, налаштовуючи параметри, надаючи текстові запити та вибираючи конкретні області зображення для модифікації;

– редагування та покращення зображень: функцію можна використовувати для редагування та покращення існуючих зображень, покращення їх якості, додавання художніх ефектів або виправлення недоліків.

Застосування image-to-image у роботі із моделями латентної дифузії:

– художнє редагування зображень: художники та дизайнери можуть використовувати функцію image-to-image для редагування та покращення своєї роботи, додавання творчих штрихів, експериментування з різними стилями та вдосконалення свого художнього бачення;

– підвищення роздільної здатності та відновлення зображень: функцію можна використовувати для збільшення роздільності зображень низької якості до вищих роздільностей, зберігаючи деталі та покращуючи чіткість. також її можна використовувати для відновлення старих чи пошкоджених зображень, вилучаючи артефакти та недоліки.

Загалом функція image-to-image у Stable Diffusion відкриває новий фронт у роботі з зображеннями та творчому вивченні. Вона дарує користувачам можливість перетворювати існуючі зображення в унікальні художні вирази, додаючи до світу цифрового мистецтва дотик магії та творчості. При подальшому розвитку технології можна очікувати ще більше інноваційних та трансформаційних застосувань цього потужного інструменту.

2.9.3. Inpainting

Функція відновлення зображень (inpainting) - це вражаючий інструмент, який дозволяє користувачам заповнювати відсутні або пошкоджені частини зображень, «безшовно» вписуючи новий контент в існуюче зображення. Дана

функція використовує процес дифузії для поступового вдосконалення області, яку потрібно відновити, керуючись контекстом навколишнього зображення та будь-якими додатковими текстовими вказівками.

Для роботи із функцією відновлення зображень спочатку виконується створення маски: користувач створює маску, яка визначає область зображення, яку потрібно відновити. Далі моделлю виконується аналіз контексту зображення навколо відзначеної області для виділення особливостей та шаблонів [40].

Після цього запускається процес відновлення: відзначена область заповнюється поступово через серію кроків дифузії. Також можна використовувати керівництво текстовими вказівками: можна надати текстові вказівки для керівництва процесом відновлення, забезпечуючи відповідність нового контенту загальному контексту та стилю зображення. Відновлена область безшовно інтегрується в існуюче зображення, створюючи природний та послідовний вигляд. Приклад використання функції Inpainting для перемальовування фрагменту оригінального зображення наведено на рис. 2.20.

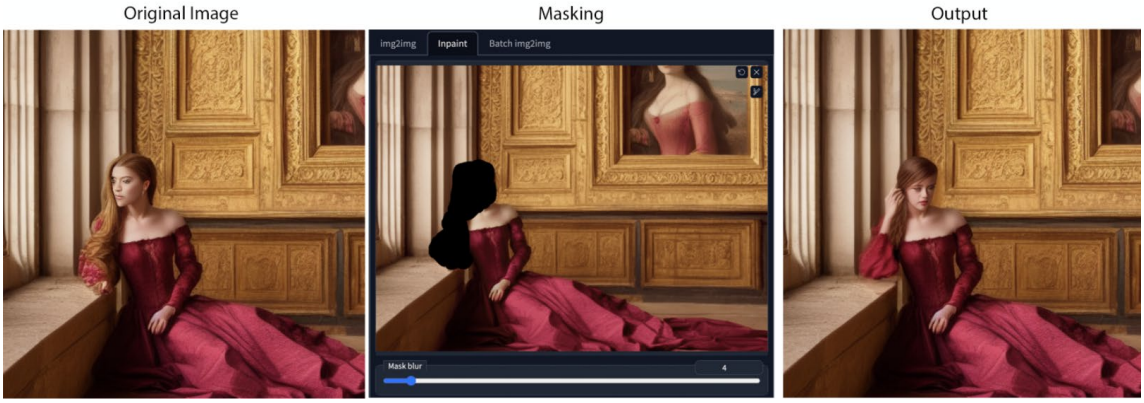

Рис. 2.20. Використання функції Inpainting для перемальовування фрагменту оригінального зображення

Основні характеристики функції Inpainting у Stable Diffusion:

– відновлення та реконструкція: функцію відновлення можна використовувати для відновлення пошкоджених чи неповних зображень, повертаючи їх до їхнього первісного стану або заповнюючи відсутні деталі;

– творче редагування та покращення: функція дозволяє користувачам творче редагувати зображення, замінюючи чи змінюючи конкретні області, зберігаючи загальну цілісність зображення;

– контекстуальна освідомленість: процес відновлення керується контекстом навколишнього зображення, забезпечуючи безшовне злиття нового контенту із існуючим зображенням;

– відновлення за допомогою тексту: текстові вказівки можуть використовуватися для надання додаткової інформації та обмежень, керуючи процесом відновлення у конкретні стилі, об'єкти чи теми.

Загалом, функція відновлення зображень представляє значний прогрес у галузі редагування та відновлення зображень. Вона дає користувачам можливість відновлювати пошкоджені зображення, покращувати існуючі фотографії та створювати унікальні художні вирази, безшовно вписуючи новий контент в існуючі зображення. З розвитком технології можна очікувати ще більше інноваційних та трансформаційних застосувань цього потужного інструменту.

2.9.4. ControlNet

ControlNet - це нейронна мережа, яка дозволяє користувачам керувати процесом генерації зображень з текстовими описами. Вона працює шляхом додавання додаткового умовного входу до моделі, який можна використовувати для керування процесом генерації зображень різними способами [41].

ControlNet інтегрується в процес генерації зображення модделю. ControlNet дозволяє модель дифузії використовувати умовний вхід для генерації зображень, які відповідають вимогам користувача. Використання даного розширення надає такі можливості як:

– керування позами живих об’єктів на зображенні. ControlNet визначає, як люди представлені на зображенні, і надає можливість контролювати їх пози;

– копіювання композиції з іншого зображення. ControlNet дозволяє використовувати інше зображення як джерело композиції, надаючи можливість точно відтворити вибрані елементи;

– генерування подібного зображення. Використовуючи ControlNet, можна контролювати параметри генерації, щоб створити зображення, схоже на вказане, додаючи деталі та адаптуючи стиль;

– перетворити чернетку в професійне зображення. ControlNet забезпечить точне керування процесом генерації, конвертуючи набросок у високоякісне та професійно виглядаюче зображення.

Користувач має можливість контролювати процес генерації зображень, налаштовуючи параметри ControlNet та вибираючи конкретні умовні входи. Наприклад, користувач може налаштувати параметри, щоб змінити стиль або деталізацію зображення. Спрощена схема інтеграції ControlNet у процес генерації зображення наведено на рис. 2.21.

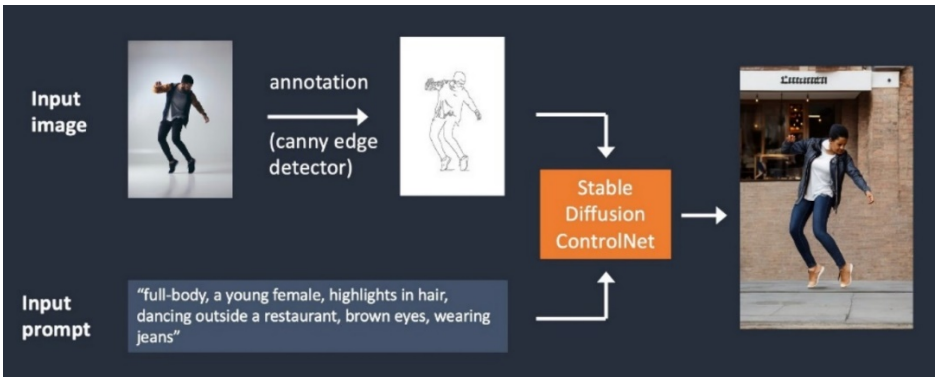

Рис. 2.21. Спрощена схема інтеграції ControlNet у процес генерації

У даному прикладі ControlNet використовується для відтворення пози людини на вхідному зображенні. Однак, хоча на вихідному зображенні і була збережена оригінальна композиція, проте це нове зображення.

Розглянемо приклад створення варіацій існуючого зображення, на прикладі птаха. На вхід нейронної мережі моделі подається зображення птаха, після чого препроцесор нейронної мережі обробляє його, визначаючи контури зображення, далі отримане при обробці зображення використовується

нейронною мережею для подальшого генерування нових зображень, які відповідатимуть вхідному. Приклад використання ControlNet для створення варіацій вхідного зображення наведено на рис. 2.22.

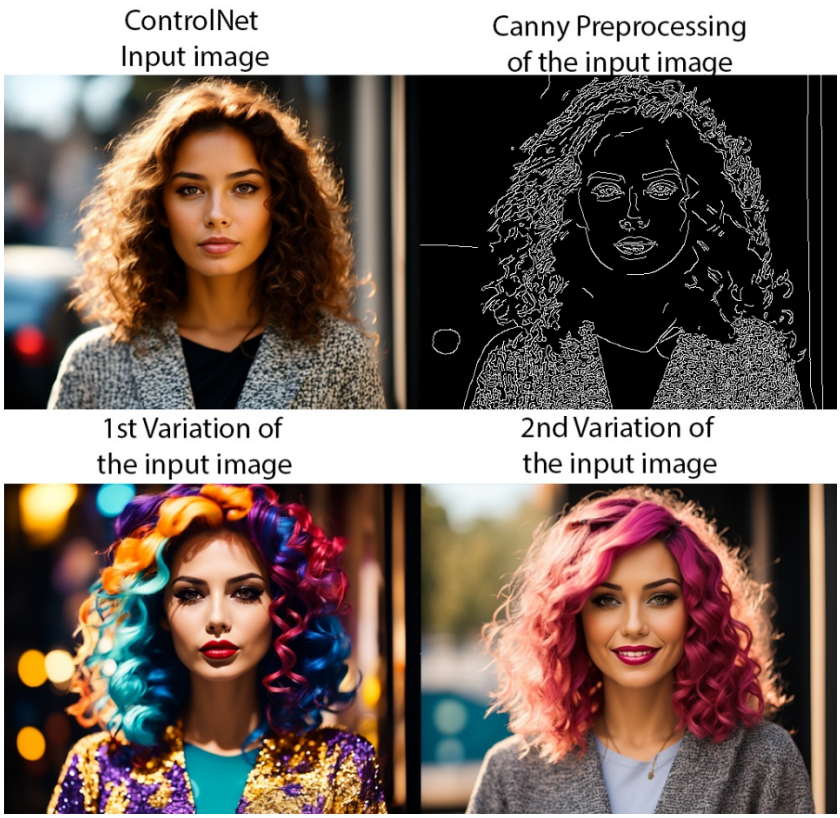

Рис. 2.22. Створення варіацій вхідного зображення, використовуючи модель ControlNet

Використання моделі ControlNet у поєднанні з моделями латентної дифузії є ефективним інструментом для керування процесом створення зображень, що дає користувачу змогу більш ретельно контролювати вихідне зображення. Наприклад, для відтворення композиції або положень на поз об’єктів на вихідному зображення.

2.10. Висновок до другого розділу

У другому розділі було теоретично розглянуто генеративні моделі, засновані на дифузії. Особливу увагу було приділено конкретному класу даних

моделей – моделям латентної дифузії. Ключовою відмінністю даного класу моделей став перенос обчислень із піксельного простору зображення у латентний простір меншої розмірності. Це зробило навчання даних моделей значно економічно- та ресурсо- ефективнішим, порівняно із вартістю навчання інших типів генеративних моделей. Введення нової архітектури також дозволило зробити використання моделей латентної дифузії масовим завдяки зниженню вимог до технічних характеристик обчислювальної машини на якій дані моделі використовуються, знизивши вимоги до обсягу відеопам'яті та обчислювальних ядер графічного ядра.

У даному розділі було розглянуто архітектуру, алгоритм процесу тренування та використання моделей латентної дифузії. Слід зазначити, що дані моделі не є монолітними структурами, вони складаються з трьох ключових компонентів, кожен з яких являє собою окрему нейронну модель:

– Variational Autocoder – нейронна модель, яка складається з двох моделей: енкодера та декодера. Енкодер використовується для створення латентного представлення вхідного зображення під час навчання або при генерації зображення на основі вхідного зображення. Декодер виконує реконструкцію початкових даних, закодованих енкодером, конвертуючи числове латентне представлення назад у піксельне зображення;

– U-Net – ключова модель у складі моделей латентної дифузії, яка виконує прогнозування та видалення шуму у вхідному представленні зображення;

– CLIP – нейронна мережа, яку було треновано для генерації латентних представлень зображень та їх текстових описів. У складі моделей латентної дифузії використовується для кодування текстового опису із запиту (Prompt) у його латентне представлення для подальшого використання у процесі генерації зображень із метою забезпечення відповідності вихідного зображення наданому текстовому опису.

Було розглянуто ключові особливості використання моделей латентної дифузії у задачах генеративного мистецтва, такі як, наприклад їх можливість

створювати зображень високої якості (моделі латентної дифузії можуть генерувати зображення, які часто неможливо відрізнити від зображень, створених людьми), відносна простота використання (моделі латентної дифузії є відносно простими у використанні, що робить їх доступними для широкого кола користувачів) та можливість генерувати різні типи зображень (моделі латентної дифузії можна використовувати для генерації різних типів зображень, включаючи реалістичні або абстракті зображення).

Також було розглянуто перелік параметрів, що використовуються під час генерації та розширень з якими можуть працювати моделі латентної дифузії під час процесу генерації зображень. Подальші дослідження будуть присвячені практичному використанню моделей латентної дифузії у задачах генеративного мистецтва із метою дослідження впливу параметрів генерації на якість вихідних зображень. Отримані результати будуть використані для формування рекомендацій, які дозволять пришвидшити та оптимізувати робочий процес при використанні моделей латентної дифузії у задачах генеративного мистецтва для отримання якісних зображень та зменшення необхідної кількості обчислювальних ресурсів.

РОЗДІЛ 3
ПРАКТИЧНЕ ВИКОРИСТАННЯ МОДЕЛЕЙ ЛАТЕНТНОЇ ДИФУЗІЇ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1. Способи практичного використання дифузійних моделей

Використання моделей дифузії вимагає певної підготовки середовища, такої як завантаження та встановлення певних бібліотек для роботи із моделями машинного навчання, наприклад:

- бібліотеки Python: NumPy, TensorFlow, Jax, JAXlib, CUDA;
- бібліотеки обробки зображень: Pillow, OpenCV, OpenImageIO;
- бібліотеки машинного навчання: Flax, Flax.linen, Haiku, Optax.

Також необхідно завантажити відповідні версії компонентів дифузійної моделі, які використовуються під час процесу навчання та роботи моделі, такі як VAE (Variational autoencoder), U-Net та CLIP.

Далі буде розглянуто можливі варіанти використання моделей латентної дифузії, від варіанту де з певних причин користувач не має змоги використовувати моделі локально, наприклад брак обчислювальної потужності або вільного місця на накопичувачі, до варіанту із локальним використанням моделей латентної дифузії.

3.1.1. Google Colab

Використання дифузійних моделей вимагає певної обчислювальної потужності, тому якщо конфігурація персонального комп’ютера користувача не дозволяє йому запустити дані моделі локально, він може використати віддаленні обчислювальні середовища, як, наприклад Google Collab. Google Colab (Google Colaboratory) це інтерактивний обчислювальний сервіс, представлений компанією Google. Він складається з середовища Jupyter Notebook, яке дозволяє запускати програмний код, написаний мовою програмування Python. Google

Colab широко застосовується у науці даних (data science), наприклад для дослідження моделей глибокого навчання [42].

Використання моделей дифузії вимагає попередньої підготовки виконавчого середовища у вигляді вибору апаратної конфігурації, встановлення необхідних дифузорів (бібліотеки, які були розроблено для роботи с дифузійними моделями, їх використання дозволяє інтегрувати різноманітні додатки для поліпшення робочого процесу та розширення функціоналу базових моделей) та формування запиту і запуск процесу генерації зображення моделлю.

Процес використання моделей дифузії у середовищі Google Colab виконується наступним чином:

3.1.1.1. Апаратне налаштування

По-перше слід переконати, що у якості апаратного прискорювача (компонента, який буде виконувати обчислення та обробку даних) обрано GPU (Graphics processing unit), бо це значно пришвидшить обчислювальний процес. Для цього потрібно у меню «Runtime» обрати пункт «Change runtime type», та обрати один із доступних варіантів, що базуються на GPU (у нашому випадку це T4 GPU). Приклад вибору апаратного прискорювача наведено на рис. 3.1.

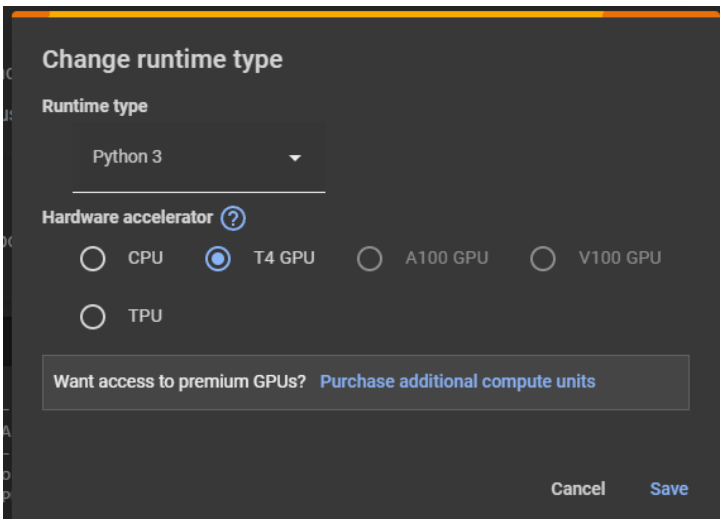

Рис. 3.1. Вибір апаратного прискорювача

Тепер можна перевірити стан обрано GPU, написавши наступну команду: !nvidia-smi. Результат перевірки стану та технічних характеристик обраного GPU наведено на рис. 3.2.

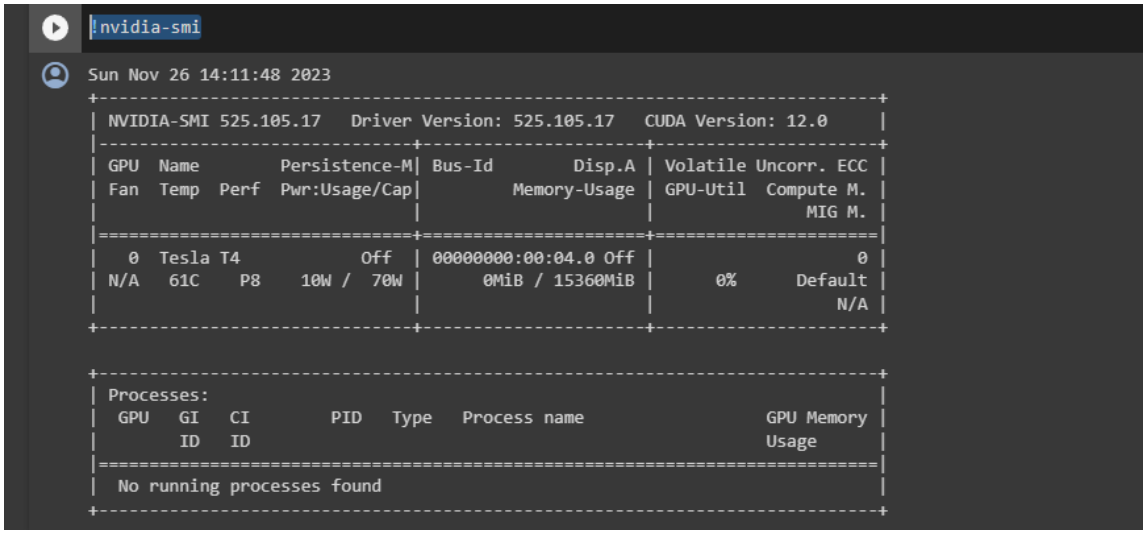

Рис. 3.2. Перевірка стану та технічних характеристик GPU

3.1.1.2. Підготовка середовища та встановлення бібліотек

Наступним кроком є встановлення бібліотек-дифузорів (diffusers) необхідних для роботи із моделями латентної дифузії. Diffusers - це компоненти латентних дифузійних моделей (LDM), які використовуються для поступового видалення шуму з латентного представлення. Цей процес називається оберненим процесом дифузії (reverse diffusion process). Для роботи можна зробити використавши готовий модуль, або зібрати його самому із наявних компонентів (відповідні треновані моделі). Для цього потрібно встановити попередньо навчені моделі:

– text_encoder: використовується для перетворення вхідного текстового запиту у вектор латентного представлення (Stable Diffusion використовує CLIP, але інші моделі дифузії можуть використовувати інші кодери, такі як BERT);

– tokenizer - токенізує текстовий опис для подальшої обробки CLIP (він повинен відповідати тому, який використовується моделлю text_encoder);

– scheduler - алгоритм планування, який використовується для поступового додавання шуму до зображення під час навчання;

– unet - модель, яка використовується для генерування латентного представлення вхідних даних та їх обробки;

– vae - модуль автокодера, який використовується для декодування латентних представлень у піксельні зображення.

Для встановлення даних компонентів потрібно ввести наступні команди:

```
import torch
torch_device = "cuda" if torch.cuda.is_available() else "cpu"
from transformers import CLIPTextModel, CLIPTokenizer
from diffusers import AutoencoderKL, UNet2DConditionModel, PNDMScheduler
vae = AutoencoderKL.from_pretrained("CompVis/stable-diffusion-v1-4", subfolder="vae")
tokenizer = CLIPTokenizer.from_pretrained("openai/clip-vit-large-patch14")
text_encoder = CLIPTextModel.from_pretrained("openai/clip-vit-large-patch14")
unet = UNet2DConditionModel.from_pretrained("CompVis/stable-diffusion-v1-4", subfolder="unet")
from diffusers import LMSDiscreteScheduler
scheduler = LMSDiscreteScheduler.from_pretrained("CompVis/stable-diffusion-v1-4", subfolder="scheduler")
vae = vae.to(torch_device)
text_encoder = text_encoder.to(torch_device)
unet = unet.to(torch_device)
```

Завантаження наведених компонентів представлено на рис. 3.3.

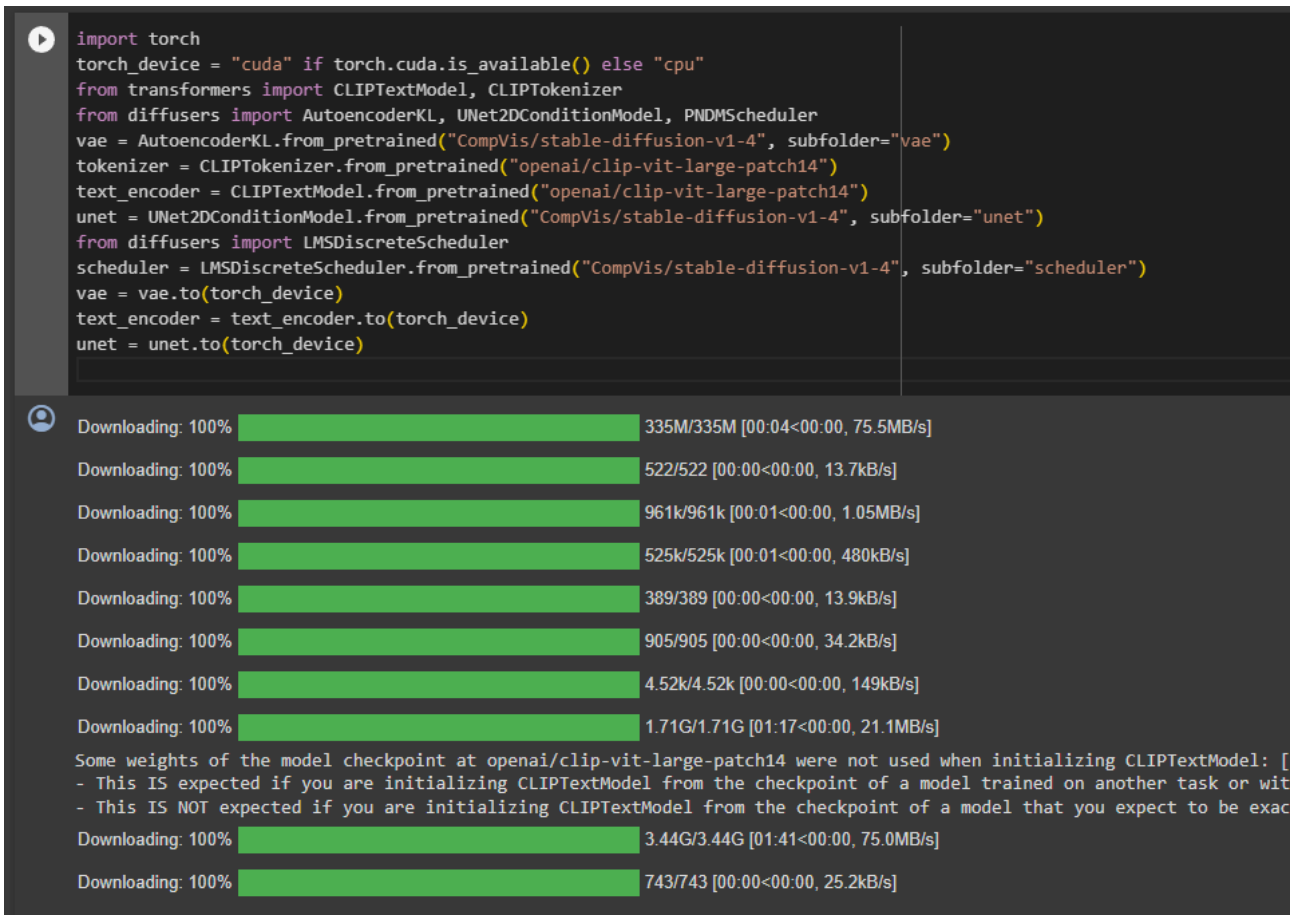

Рис. 3.3. Завантаження необхідних компонентів

Усі завантажені компоненти були треновані у складі моделі Stable Diffusion 1.4. У даному прикладі використовується модель Stable Diffusion версії 1.4, проте існують і інші наявні версії даної дифузійної моделі. Вони відрізняються кількістю даних, на яких вони були треновані, їх розмірністю, а також якістю та цілями навчання моделей. Обрана модель впливає на якість та рекомендований розмір генерованого зображення (моделі до версії Stable Diffusion 2.0 навчалися на наборах зображень розміром 512x512 пікселів, версії, починаючи з версії 2.0 навчалися на зображеннях розміром 768x768 пікселів).

3.1.1.3. Підготовка параметрів генерації

Тепер, коли середовище було підготовано, можна перейти до генерації зображень із використанням усіх наведених компонентів. Спочатку необхідно сформувати список параметрів генерації, такі як текстовий запит (Prompt),

розмір зображення (height та width), кількість кроків виконання (num_inference_steps), відповідність текстовому запиту (guidance_scale), початкове значення шуму (seed) та розмір пакунку (batch_size).

```
prompt = ["a photograph of an astronaut riding a horse"]
height = 512 # default height of Stable Diffusion
width = 512  # default width of Stable Diffusion
num_inference_steps = 100   # Number of denoising steps
guidance_scale = 7.5          # Scale for classifier-free guidance
generator = torch.manual_seed(32)   # Seed generator to create the inital latent noise
batch_size = 1
```

Спочатку сформуємо латентне представлення текстового запиту. Під час генерації дане представлення буде використано моделлю U-Net для забезпечення відповідності (conditioning) генерованого зображення текстовому опису із запиту. Також нам потрібно сформувати unconditional представлення, яке також буде використано моделлю U-Net під час генерації. Обидва представлення повинні мати однакову розмірність. Для роботи алгоритму генерації, модель потребує два представлення: одне, яке відповідає текстовому запиту, та друге – пусте, яке буде використано для генерації unconditional-представлень.

```
text_input = tokenizer(prompt, padding="max_length", max_length=tokenizer.model_max_length, truncation=True, return_tensors="pt")
with torch.no_grad():
  text_embeddings = text_encoder(text_input.input_ids.to(torch_device))[0]
 max_length = text_input.input_ids.shape[-1]
uncond_input = tokenizer(
    [""] * batch_size, padding="max_length", max_length=max_length, return_tensors="pt"
)
with torch.no_grad():
  uncond_embeddings = text_encoder(uncond_input.input_ids.to(torch_device))[0]
```

```
text_embeddings = torch.cat([uncond_embeddings, text_embeddings])
```

Наступним кроком є формування початкового тензору шуму.

```
latents = torch.randn(
  (batch_size, unet.in_channels, height // 8, width // 8),
  generator=generator,
)latents = latents.to(torch_device)
```

Даний тензор має розмірність 4x64x64. Після завершення кроків обробки, декодер VAE перетворить дане представлення у кінцеве зображення у його піксельній формі.

Далі ініціалізуємо планувальник за допомогою обраного нами num_inference_steps. Це обчислить значення sigmas та точні часові кроки, які будуть використовуватися під час процесу видалення шуму. Планувальник K-LMS повинен помножити latents на свої значення sigma.

```
scheduler.set_timesteps(num_inference_steps)
latents = latents * scheduler.init_noise_sigma
```

Цикл очищення шуму з латентного представлення. Та формування нового латентного представлення.

```
from tqdm.auto import tqdm
from torch import autocast
for t in tqdm(scheduler.timesteps):
  # expand the latents if we are doing classifier-free guidance to avoid doing two forward passes.
  latent_model_input = torch.cat([latents] * 2)
                    latent_model_input = scheduler.scale_model_input(latent_model_input, t)
  # predict the noise residual
  with torch.no_grad():
          noise_pred = unet(latent_model_input, t, encoder_hidden_states=text_embeddings).sample
  # perform guidance
  noise_pred_uncond, noise_pred_text = noise_pred.chunk(2)
    noise_pred = noise_pred_uncond + guidance_scale * (noise_pred_text - noise_pred_uncond)
```

```
    # compute the previous noisy sample x_t -> x_t-1
    latents = scheduler.step(noise_pred, t, latents).prev_sample
```

Тепер використаємо VAE Decoder, щоб сформувати зображення з отриманого латентного представлення. Та сформуємо на основі отриманих даних фінальне зображення.

```
latents = 1 / 0.18215 * latents
with torch.no_grad():
  image = vae.decode(latents).sample
image = (image / 2 + 0.5).clamp(0, 1)
image = image.detach().cpu().permute(0, 2, 3, 1).numpy()
images = (image * 255).round().astype("uint8")
pil_images = [Image.fromarray(image) for image in images]
pil_images[0]
```

Результатом конвертації латентного представлення у піксельний простір є вихідне зображення, наведене на рис. 3.4.

Рис. 3.4. Перегляд вихідного зображення

Однак, наразі Google заборонив роботу з дифузійними моделями у рамках безкоштовного користувальницького плану. Використання дифузійних моделей вимагає придбання відповідної підписки, яка надасть користувачеві певний обсяг обчислювальної потужності (compute units).

3.1.2. Онлайн-сервіси

Також є онлайн-сервіси, які використовують моделі, засновані на латентній дифузії. Більшість даних сервісів надає змогу безкоштовно згенерувати певну кількість зображень із обмеженим налаштуванням, наприклад користувач не може змінити значення розміру, самостійно обрати навчену модель, яка б краще відповідала його потребам, встановити необхідні йому розширення.

Дані сервіси є зручним інструментом для первинного та поверхневого ознайомлення із можливостями моделей латентної дифузії, але вони не дають користувачеві отримати повного спектру досвіду взаємодії з ними. А варіанти сервісів, які все ж надають користувачу повний спектр можливих налаштувань, зазвичай є платними, що також робить їх не самим оптимальним вибором, якщо стоїть така ціль як довгострокове практичне дослідження та експерименти із налаштуванням параметрів генерації, їх вплив на вихідні зображення та інше. Також використання сторонніх сервісів накладає значні обмеження на те, які розширення користувач може використовувати (абсолютну більшість з доступних розширень необхідно встановлювати самому, коли онлайн-сервіси не надають такої змоги).

3.1.3. Stable Diffusion WebUI

Тепер розглянемо гарантовано безкоштовний варіант використання дифузійних моделей, який надає повний спектр можливостей та налаштувань, доступних при роботі із моделями латентної дифузії – локальне встановлення. Можна по аналогії із використанням Google Colab окремо завантажувати усі компоненти середовища, необхідні для роботи із моделями та їх розширеннями, проте ентузіастами команди Automatic1111 (та іншими користувачами, які долучилися до процесу розробки) було розроблено зручне програмне середовище – Stable Diffusion WebUI, яке представляє собою графічний інтерфейс для зручної роботи із моделями латентної дифузії та їх розширеннями.

Використання даного застосунку не є обов'язковим, проте це значно полегшує та пришвидшує робочий процес [43]. Дане розширення дозволяє користувачу зручно використовувати наявні розширення та встановлювати нові, «безшовно» інтегруючи їх у робочий процес для взаємодії із дифузійною моделлю. Приклад графічного інтерфейсу Stable Diffusion WebUI наведено на рис. 3.5.

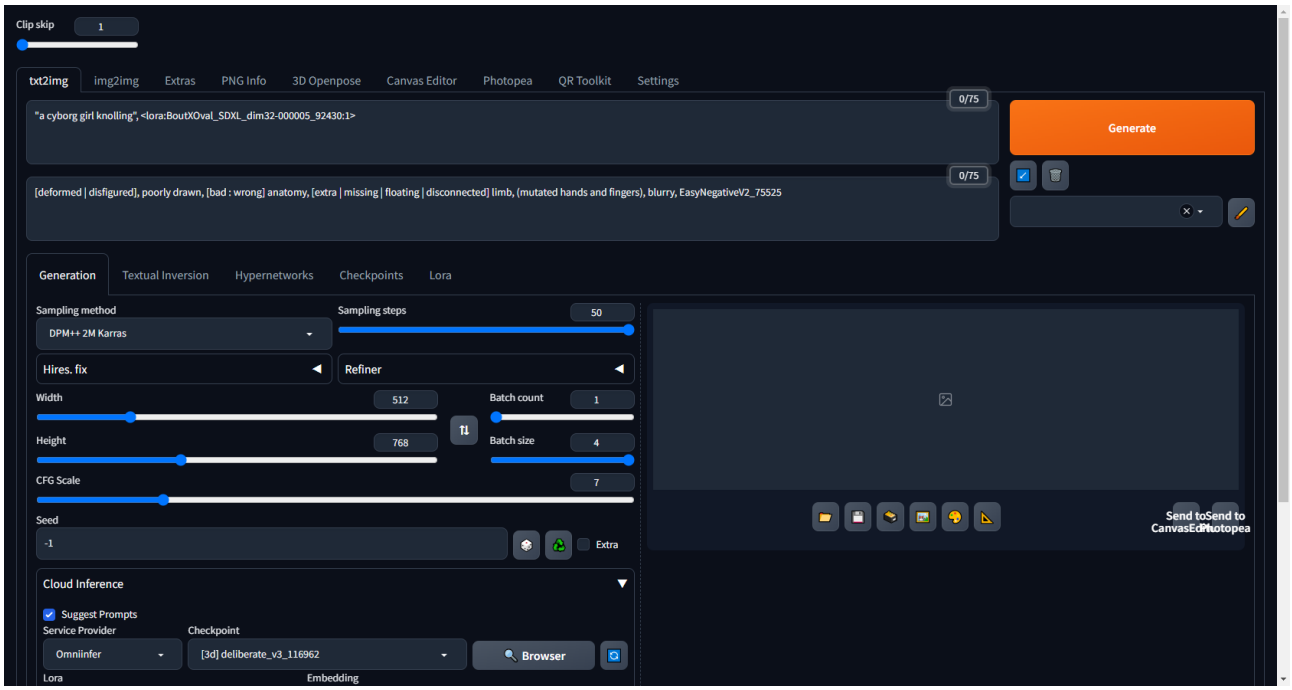

Рис. 3.5. Графічний інтерфейс Stable Diffusion WebUI

Використання даної графічної оболонки дозволяє користувачу швидко та зручно перейти до використання та дослідження моделей латентної дифузії, зосередившись саме на практичному аспекті, замість написання відповідного коду для формування параметрів генерації та використання доступних розширень. Тому, для зручності та систематизації наступних практичних досліджень моделей латентної дифузії буде використано Stable Diffusion WebUI.

3.2. Експериментальне дослідження та аналіз результатів

У рамках експериментального дослідження буде проведено генерацію зображень із використанням різних параметрів генерації для дослідження їх

впливу на вихідне зображення. Також буде сформовано та розв’язано перелік типових завдань, які постають перед користувачем при роботі із моделями латентної дифузії у сфері генеративного мистецтва:

– генерація зображень на основі текстового опису;

– генерація зображень на основі вхідного зображення;

– створення варіацій вхідного зображення, такі як відтворення композиції вхідного зображення, керування позами та положенням об’єктів, зміна стилістики вхідного зображення;

– пост-обробка отриманих зображень із метою підвищення їх якості, деталізації та роздільної здатності.

Уявимо, що потенційний користувач бажає якомога якісніше та точніше згенерувати наступні зображення:

– фотографія молодої дівчини, яка гуляє по вечірнім вулицям у Токіо. Дівчина повинна мати рожеве волосся середньої довжини, синю джинсову куртку, чорні джинсові штани. Фон – типова вулиця японського мегаполісу, навколо дівчини є прохожі люди, вітрини та вивіски мають неонове освітлення.

– фотографія живописного пейзажу: зелений луг, за яким слідують високі гори, верхівки яких вкриті снігом. Час фотографії – вечір, червоно-рожеве небо на заході сонця.

3.2.1. Дослідження впливу параметрів генерації

3.2.1.1. Вплив обраної моделі латентної дифузії

Вибір попередньо тренованої моделі відіграє ключову роль у процесі генерації зображень, адже якість навчання та набір даних, на яких було треновано обрану модель, буде впливати на стилістику, деталізацію та якість вихідного зображення. Обрана модель повинна відповідати потребам користувача до бажаних зображень, адже різні моделі було треновано для створення певних типів та стилів зображень: фотореалістичні зображення,

наприклад людей, пейзажів чи об’єктів, ілюстрації та малюнки, абстрактні зображення [44].

Наразі є два крупних онлайн-агрегатори моделей латентної дифузії: Civit та huggingface.co, які дозволяють користувачам безкоштовно ознайомитись із наявними на ресурсі моделями (переглянути специфіку та області їх використання у контексті генерації зображень: фотореалістичні зображення, фотографії людей або тварин, абстрактні зображення, ілюстрації тощо) та завантажити їх. Приклад перегляду наявних на ресурсі Civitai.com тренованих моделей для генерації зображень наведено на рис. 3.6.

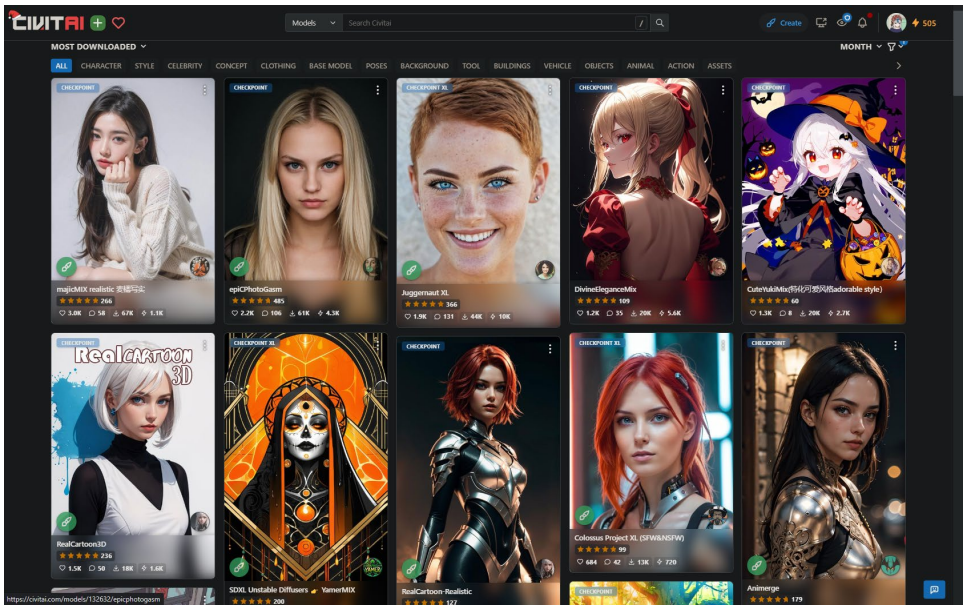

Рис. 3.6. Перегляд наявних на ресурсі Civitai.com тренованих моделей

Є множина моделей, які доступні до завантаження у мережі, проте вони не є повністю оригінальними, насправді, вони є результатом до-навчання оригінальних моделей Stable Diffusion, які використовуються у якості базової моделі (Base model). Для того, щоб отримати нову модель, виконується додаткове тренування (Additional training) базової моделі на наборі зображень певної тематики. Наприклад, якщо провести додаткове тренування базової модель на наборі даних раритетних автомобілів, то результатом буде модель для генерації зображень раритетних авто.

Розглянемо приклад генерації зображень при аналогічних параметрах генерації, але із використанням моделей, тренованих для різних цілей:

– cetusMix_v4 – створення ілюстрацій персонажів;

– Deliberate_v2 –створення реалістичних зображень;

– Deliberate_v3 - створення реалістичних зображень

– disneyPixarCartoon_v10 - створення ілюстрацій у стилі мультфільмів Disney та Pixar;

– epicrealism_newCentury - створення реалістичних зображень людей;

– landscapesupermix_v21 – створення зображень пейзажів та архітектури;

– marvelousdangeonsnewv30 – створення ілюстрацій;

Використання різних моделей генерації, при аналогічних параметрах генерації наведено на рис. 3.7.

Рис. 3.7. Вплив використання моделей із різною стилістикою

Можна побачити, що різні моделі дають надають різну стилістику вихідним зображенням. Наприклад, моделі, такі як, disneyPixarCartoon та cetusMix, які було треновано для створення мультфільмів та ілюстрацій, надають зображення у даній стилістиці, при їх використанні буде неможливо виконати створення фотореалістичних зображень, та навпаки, моделі для створення реалістичних зображень, такі як, Deliberate, epicrealism або landscapesupermix не можуть бути використані для створення якісних ілюстрацій. Існує множина тренованих моделей, які було треновано для створення специфічних об’єктів та місць, дані моделі можуть використані відповідно до побажань користувача.

3.2.1.2. Вплив текстового опису (Prompt)

Добре сформований текстовий опис (Prompt) є невід'ємною складовою якісно-згенерованого зображення, яке відповідає побажанням та потребам користувача, тому слід розуміти як правильно його сформувати. Не має однозначно «правильного» текстового опису, проте є загальні рекомендації, які дозволяють зробити процес генерації більш контрольованим та послідовним.

Розглянемо вплив якості написання текстового запиту на фінальне зображення на конкретних прикладах. Для дослідження впливу саме текстового опису під час генерації зображень усі налаштування, окрім текстового опису будуть залишатися незмінними.

Спробуємо написати текстовий опис бажаного зображення із різними рівнями конкретики, деталізації та точності опису, та перевіримо результати виконання кожного з запитів.

Сформовані варіанти можливих запитів:

- a young lady
- A photo of a young lady
- A photo of a pretty young lady with pink hair
- A photo of a pretty young lady with pink hair + evening + in the middle of a cozy street + Tokyo + cold neon street lights + denim jacket, black denim jeans + public, people around.

Тепер проведемо генерування за кожним із запитів та об'єднаємо отримані результати у пари текстовий опис – зображення. Варіанти текстових описів та зображення, згенеровані на їх основі наведено на рис. 3.8.

Рис. 3.8. Варіанти модифікації текстових описів та зображень, згенерованих на їх основі

Таким чином, для першого зображення текстовий опис має бути наступним: A photo of a young pretty lady with pink hair, cowboy shot, evening, in the middle of a cozy street, Tokyo, cold neon street lights, denim jacket, black denim jeans, public, people around.

Тепер спробуємо варіанти написання запиту для другого зображення.

– landscape

– photo of a lanscape + lush green valley + green meadow + followed by a snowy mountain + photorealistic + evening + sunset + pink and red sky + naturalism.

Варіанти текстових описів та зображення, згенеровані на їх основі наведено на рис. 3.9.

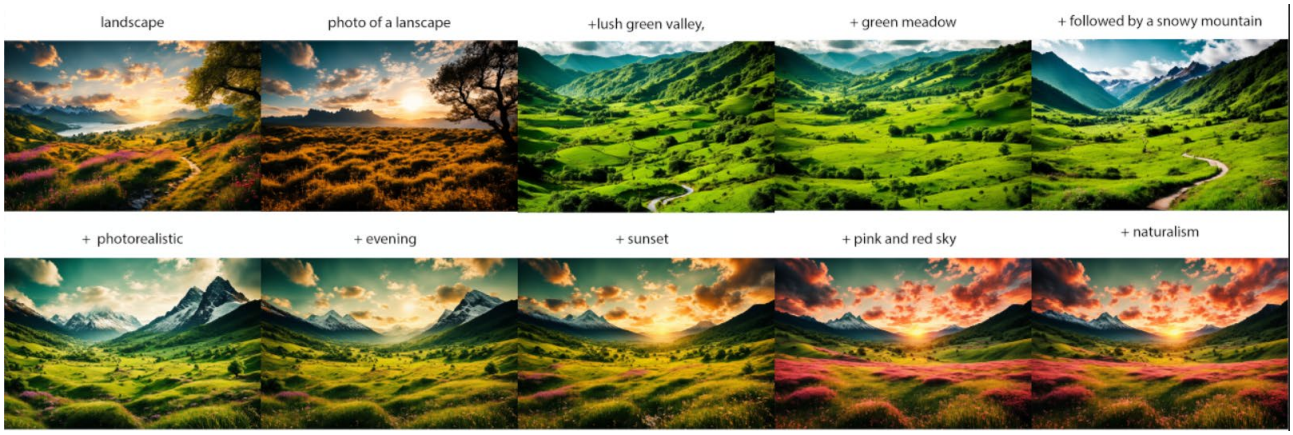

Рис. 3.9. Варіанти текстових описів та зображення, згенеровані на їх основі

3.2.1.3. Вплив кількості кроків обробки (Sampling steps)

Наступним важливим параметром є кількість кроків обробки (sampling steps). Модель створює зображення, починаючи з полотна, повного шуму, і поступово зменшує шум, щоб досягти кінцевого результату. Цей параметр контролює кількість цих кроків зменшення шуму. Зазвичай, чим вище, тим краще, але до певного ступеня. За замовчуванням використовуємо 25 кроків, яких має бути достатньо для генерування будь-якого типу зображення. Раніше для отримання якісних зображень за допомогою старих семплерів, таких як LMS, часто використовували велику кількість кроків, іноді навіть 100 або 150. Однак з появою нових, більш швидких та ефективних семплерів, таких як семплери групи DPM Solver++, потреба в такій кількості кроків пропала. Використання великої кількості кроків з цими семплерами, як правило, не призводить до покращення якості зображення, а лише збільшує час генерації та навантаження на графічний процесор. Тому в більшості випадків для отримання якісних зображень за допомогою сучасних алгоритмів очищення достатньо використовувати близько 25 кроків.

Приклад впливу кількості кроків обробки на вихідне зображення наведено на рис. 3.10.

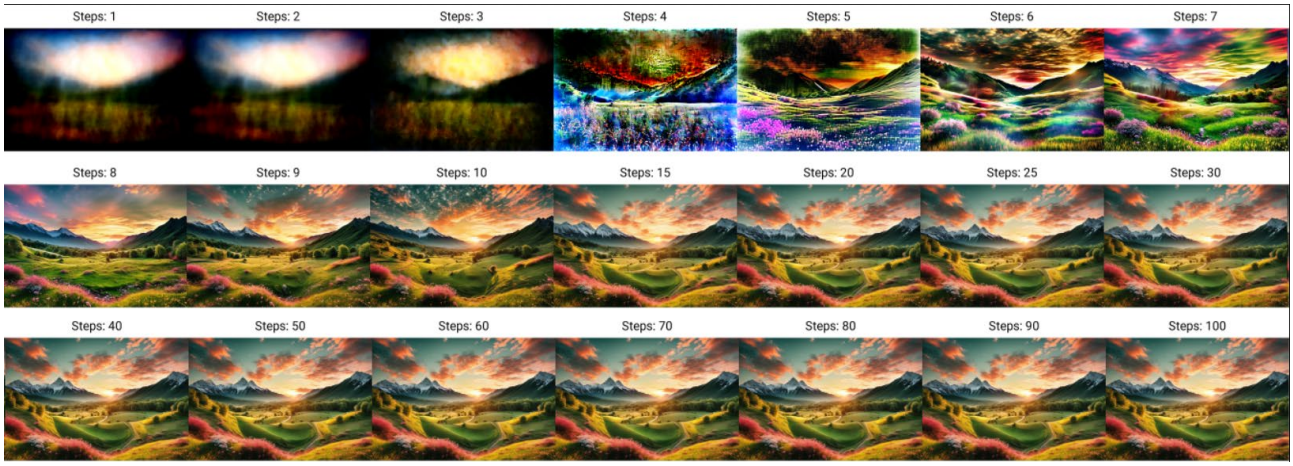

Рис. 3.10. Вплив кількості кроків обробки на вихідне зображення

На наведеному прикладі можна побачити, що використання занадто малої кількість кроків вибірки призводить до створення зображень, які не відповідають тестовому опису та не є придатними до використання. Використання ж занадто великої кількості кроків призводить до того, що зображення починає перенасищатися зайвими деталями, яких не було у текстовому описі, до того ж виконання такої кількості кроків обробки вимагає значно більше часу та обчислювальної потужності. Натомість, оптимальним є діапазон 20-30 кроків обробки, на якому вихідні зображення найкраще зберігають баланс між відповідністю текстовому запиту і якістю та необхідним на обробку часом та обчислювальною потужністю.

3.2.1.4. Вплив методу очищення шуму (Sampling methods)

Метод очищення шуму (Sampling methods) це алгоритм, який використовується під час генерації зображення дифузійними моделями на етапі обробки вхідного шуму. Обраний метод очищення є математичною функцією, та відповідно до її властивостей, вибір методу очищення впливає на вихідне зображення та швидкість генерації.

Алгоритми роботи методів мають певні відмінності, проте загально їх можна розділити на 2 групи:

Стохастичні – методи вибірки, що додають на кожному кроці певну частку випадкового шуму до латентного представлення. Це дозволяє збільшити варіативність та деталізацію вихідних зображень, адже для моделей дифузії таке використання нових ймовірностей є гарним джерелом нових даних, які можуть бути оброблені та репрезентовані по-новому. Однак, у випадку використання методів із даної групи слід очікувати, що зображення згенеровані на ідентичних параметрах не будуть співпадати (саме через додавання нового випадкового проміжного шуму на кожному етапі).

Послідовні – це методи вибірки, які не додають нового шуму, а лише реалізують процес видалення «прогнозованого» моделлю U-Net шуму від

поточного представлення. Їх використання потребує менше часу та обчислювальної потужності, при цьому дозволяючи отримувати стабільні та відтворювані результати.

Вплив методів вибірки на вихідні зображення для першого текстового опису наведено на рис. 3.11.

Рис. 3.11. Вплив методів вибірки на вихідні зображення

Вплив методів вибірки на вихідні зображення для другого текстового опису наведено на рис. 3.12.

Рис. 3.12. Вплив методів вибірки на вихідні зображення

Для зображень, які не містять занадто багато семантично-значущих деталей, наприклад як фотографії пейзажів, вибір методу вибірки не є занадто критичним, однак для фотографій із більш значущим семантичним значенням та деталізацією, як наприклад зображення людей, які мають обличчя та руки, вибір методу вибірки відіграє більш значну роль. Використання різних методів призводить до більшої варіативності вихідних зображень відносно текстового опису, наприклад, як методи вибірки Euler a, DPM2 a, DPM++ 2S a, які вносять на кожному кроці обробки частину випадкового шуму, призводячи до того, що вихідне зображення відрізняється від заданого текстового опису, наприклад, як зміна кольору верхнього одягу дівчини на наведених прикладах. Підсумовуючи отримані результати, будь-який метод із групи DMP++ 2S (окрім предківської версії DPM++ 2S a) є прийнятним для створення зображень людей.

3.2.1.5. Вплив параметра CFG Scale

Параметр CFG (Classifier Free Guidance) Scale визначає те, наскільки текстовий опис із запиту буде впливати на процес генерації, спрощено це можна описати як «вагу» текстового опису. Значення даного параметра дозволяє встановити баланс між креативністю та відповідністю текстовому опису під генерації. Нижчі значення CFG Scale призводять до більшої варіативності вихідних зображень відносно текстового опису, більші значення CFG Scale призводять до того, що генероване зображення точніше відповідає текстовому опису, проте, як буде наведено на прикладах, можна побачити, що занадто високі значення (зазвичай вище 15) призводять до деформації зображення та появи артефактів.

Введення метода CFG Scale зробило можливим процес розробки дифузійних моделей, адже попередні версії генеративних моделей, засновані на GAN вимагали використання у своєму складі двох окремих нейронних моделей: генератора та дискримінатора. Таким чином, для реалізації процесу генерації потрібно було навчити одразу дві моделі та потім та само одночасно їх

використовувати, що значно збільшувало вимоги до обчислювальних можливостей апаратного пристрою.

У свою чергу поява методу CFG Scale дозволила перейти до створення генеративних моделей, які більше не потребували дискримінатор, та дозволяли інтегрувати інші моделі, які давали більший контроль над процесом генерації. Вплив параметра SFG Scale на вихідні зображення для кожного з наведених раніше текстових описів наведено на рис. 3.13.

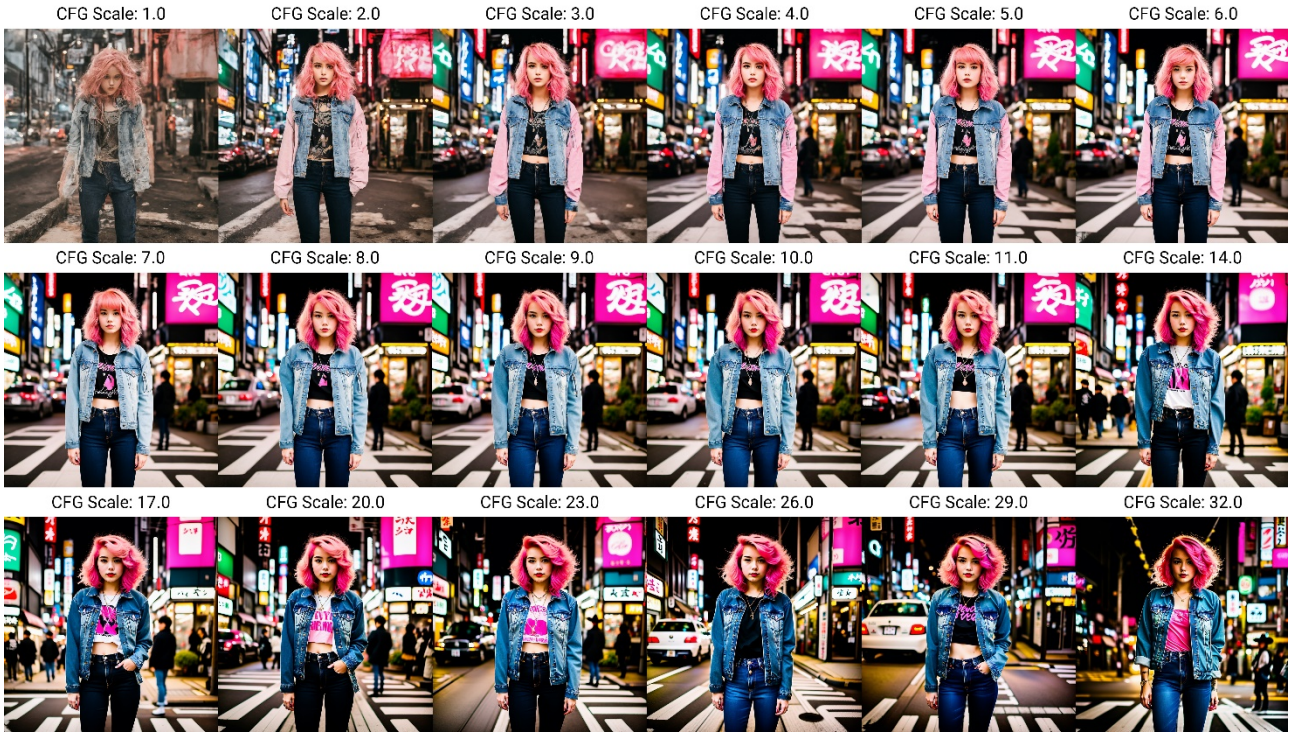

Рис. 3.13. Вплив параметра SFG Scale на вихідні зображення

Як можна побачити з отриманих зображень, значення параметра CFG Scale впливає на відповідність вихідного зображення наданому текстовому опису. Однак, використання занадто малих (нижче 5) та занадто великих (більше 14) значень даного параметру можуть призводити до нечіткості зображення, або його сильного спотворення через появу ненатуральної кольорової палітри, що робить зображення менш схожим на реальну фотографію, або взагалі, зображення, зроблене людиною.

3.2.1.6. Вплив розміру зображення (Image size)

Вибір розміру зображення зазвичай залежить від конкретної моделі дифузії, яку використовують під час генерації. Ключовим показником, на який слід орієнтуватися, є розмір зображень із тренувального набору даних на яких було треновано обрану модель. Більшість наявних моделей було треновано на наборах зображень із розмірністю 512x512 пікселів, тому оптимальним є використання даного розміру, або таким чином, щоб хоча б одна зі сторін мала розмір 512 пікселів. Тож, обирати розмір зображення бажано відповідно до базової моделі на якій було додатково треновано модель, яка використовується.

Тепер можна перейти до практичної перевірки впливу різних параметрів розміру на вихідне зображення. Встановимо різні співвідношення та розміри зображення, таким чином, щоб частина відповідала розмірності використаній при тренуванні моделі, а інша частина – ні. Результат використання різних розмірів та співвідношень під час генерації наведено на рис. 3.14.

Рис. 3.14. Вплив різних параметрів розмірів та співвідношень на вихідні зображення

Можна побачити, що на зображеннях, які зберігали відповідність розміру зображень із тренувального набору моделі, не має структурних спотворень, та навпаки: зображення, які мали занадто великий розмір мають певні аномалії, починаючи від простого дублювання об’єктів до спотворення людської анатомії.

Також слід пам’ятати, що навіть незначна зміна розміру зображення, навіть при ідентичних налаштуваннях усіх інших параметрів, цілком змінить вихідне зображення. Використання розмірів, що не узгоджуються із розмірами зображень з тренувального набору моделі може викликати дублювання та спотворення об’єктів на вихідному зображенні.

3.2.1.7. Вплив початкового шуму (Seed)

Параметр seed визначає шаблон початкового шуму зображення. Використання одно й того ж значення seed дозволяє відтворити зображення при зберіганні решти налаштувань тими ж самими. Зміна значення seed призведе до генерації абсолютно нового зображення. Під час генерації зображень можна експериментувати із різними значеннями параметра seed, доки ви не отримаєте зображення, яке влаштовує вас за композицією. Далі, залишаючи те ж значення seed можна вносити зміни у текстовий запит, та отримувати зображення, які будуть дуже близькі за структурою, проте будуть містити відмінності, внесені у текстовий опис. Зміна значення seed може спричинити структурні зміни у вихідному зображенні, наприклад, такі як, зміна розташування та розмірів об’єктів, їх деталей та кольорів, тощо. Для перевірки впливу постійності зображення при зберіганні значення параметра seed сформуємо 3 групи з 4 зображень, кожна груп буде використовувати одне й те ж значення seed, але різні запити. Вплив різних значень параметра seed на вихідні зображення наведено на рис. 3.15.

Рис. 3.15. Вплив різних значень параметра seed на вихідні зображення

Можна побачити, що навіть незначна зміна значення параметра seed призводить до створення нових зображень. Проте, як і очікувалося, при незначних змінах у текстовому описі, зображення створенні із використанням однакового значення параметра seed зберігають схожу структуру. Наприклад, якщо зміна у текстовому описі незначна, як у другій групі, коли виконується зміна малої деталі зображення, у даному прикладі – волосся, загальна структура зображень із однаковими значеннями seed співпадає: розташування та пози об’єкта на зображенні мають багато спільного. Приклад з третьої групи значно відрізняється, адже запит було змінено таким чином, що з нього пропав опис штанів, які були у 1 та 2 групі. Це призвело до того, що модель, не маючи вказівку про створення штанів, змінила положення дівчини у кадрі таким чином, щоб воно більше співпадало із запитом (у описі є біла сорочка, але немає штанів).

3.2.1.8. Вплив Negative prompt

Наступним розглянемо такий параметр як Negative prompt. Даний параметр визначає які деталі зображення моделі не потрібно додавати до зображення. Даний параметр дозволяє запобігти появі на зображенні небажаних

деталей, наприклад, при створенні зображень людей потрібно запобічгти появі деформації чи спотворенню облич, рук та загальної анатомії людей. Проведемо серію експериментів із текстовим описом, сформованим у пункті 3.2.2.2. та розділимо генеровані зображення на три групи:

– із використанням повного negative prompt, який включає в себе запобігання деформацій та спотворення людської анатомії;

– взагалі без використання negative prompt;

– без запобігання деформацій та спотворення людської анатомії, проте із запобіганню певних деталей: приберемо темряву (ніч), людей, машини та знаки дорожнього руху, та заборонимо моделі генерувати верхній одяг блакитного кольору.

Приклад впливу використання різних варіантів формування negative prompt на вихідні зображення наведено на рис. 3.16.

Рис. 3.16. Вплив різних варіантів negative prompt на вихідні зображення

Аналізуючи отримані результати, можна зробити висновок про вплив використання negative prompt на вихідні зображення:

– використаний у першій групі negative prompt помітно покращує людську анатомію та запобігає значному спотворенню облич на вихідних зображення. Даний negative prompt є універсальним для базової корекції людської анатомії;

– у випадку коли не було застосованого ніякого опису у negative prompt можна побачити деформацію у анатомії та спотворення облич. Отримані результати мають ненатуральний вигляд та не мають бажаної якості та реалістичності, як того вимагає фотографія людини;

– у третій групі можна побачити часткову відповідність negative prompt, наприклад, зображення стали менш темними, та більше відповідають фотографіям, зроблений у денний час, також стало помітно менше людей та машин на фоні, а верхній одяг отримав більше варіацій таким чином, щоб не бути «блакитним», як це було описано у negative prompt. Проте можна помітити, що обличчя та руки піддалися певній деформації та більше не виглядають натурально. Отримані зображення не цілком співпадають із описом, наданим у negative prompt через те, що вони потрапляють у колізію із текстовим описом із Prompt, наприклад колір куртки. Це пов’язано з тим, що під час навчання моделі на зображеннях, які містили у описі «denim jacket» більшість прикладів містили «denim jacket» саме блакитного кольору. Для виправлення даної проблеми потрібно виправити Prompt таким чином, щоб він не вказував так явно на матеріал з якого зроблено куртку (denim).

– Приклад виправлених Prompt та Negative prompt:

– A photo of a young pretty lady with pink hair, cowboy shot, evening, in the middle of a street, Tokyo, cold neon street lights, jacket, black jeans;

– (blurry face:1.2), [deformed | disfigured], 3d, render, poorly drawn, [bad : wrong] anatomy, [extra | missing | floating | disconnected] limb, (mutated hands and fingers), (blue jacket:1.4), (night:1.3), (darkness:1.3), (public:1.3), people around, (cars:1.4), (street signs:1.3), road signs, low quality, worst quality.

Результат використання нових Prompt та Negative prompt наведено на рис. 3.17.

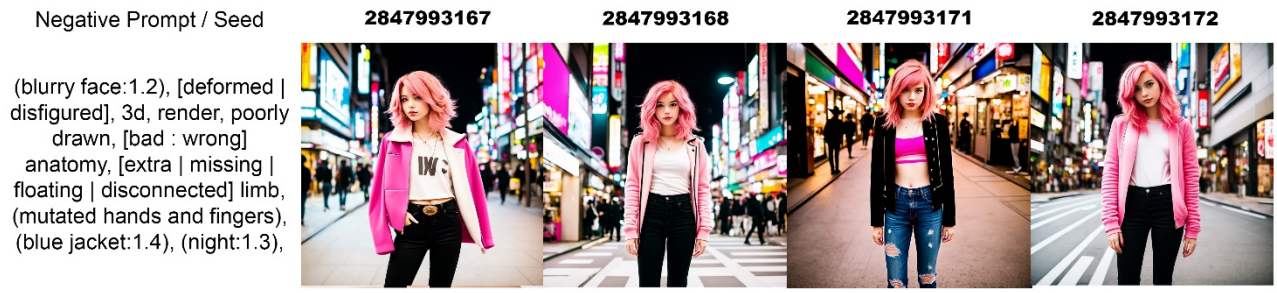

Рис. 3.17. Результат використання нових Prompt та Negative prompt

Можна побачити, що корегування Prompt та Negative prompt дозволило краще узгодити їх сумісне використання, прибравши зіткнення контрарних текстових описів у запитах. Наприклад, було отримано зображення із покращеною анатомією людей на зображеннях та змінено колір куртки.

3.2.1.9. Denoising strength

Значення параметру Denoising strength визначає те, наскільки багато шуму буде додано до представлення вхідного зображення. Даний параметр приймає значення від 0 до 1, де 0 визначає, що випадкового шуму зовсім не буде додано, а 1 – що вхідне представлення зображення буде цілком замінено на випадковий шум. Проміжні значення між 0 та 1 лінійно визначають кількість доданого випадкового шуму до вхідного представлення. Зазвичай значення нижче 0,50 дозволяють зберегти композицію близьку до вхідного зображення, а значення вище 0,50 сприяють створенню нових варіацій вхідного зображення.

Параметр denoising strength використовується у всіх режимах роботи та розширеннях моделей латентної дифузії, які використовують зображення у якості вхідних даних для процесу генерації, наприклад режим роботи image-to-image, коли нові зображення генеруються на основі вхідного зображення, та розширення, наприклад, такі як hires.fix, який використовую інформацію з вхідного зображення для створення його версій більшої якості та розмірності.

Дослідимо вплив різних значень denoising strength у діапазоні [0..1] із кроком 0.1 на вихідні зображення, приклад впливу різних значень параметра denoising strength наведено на рис. 3.18.

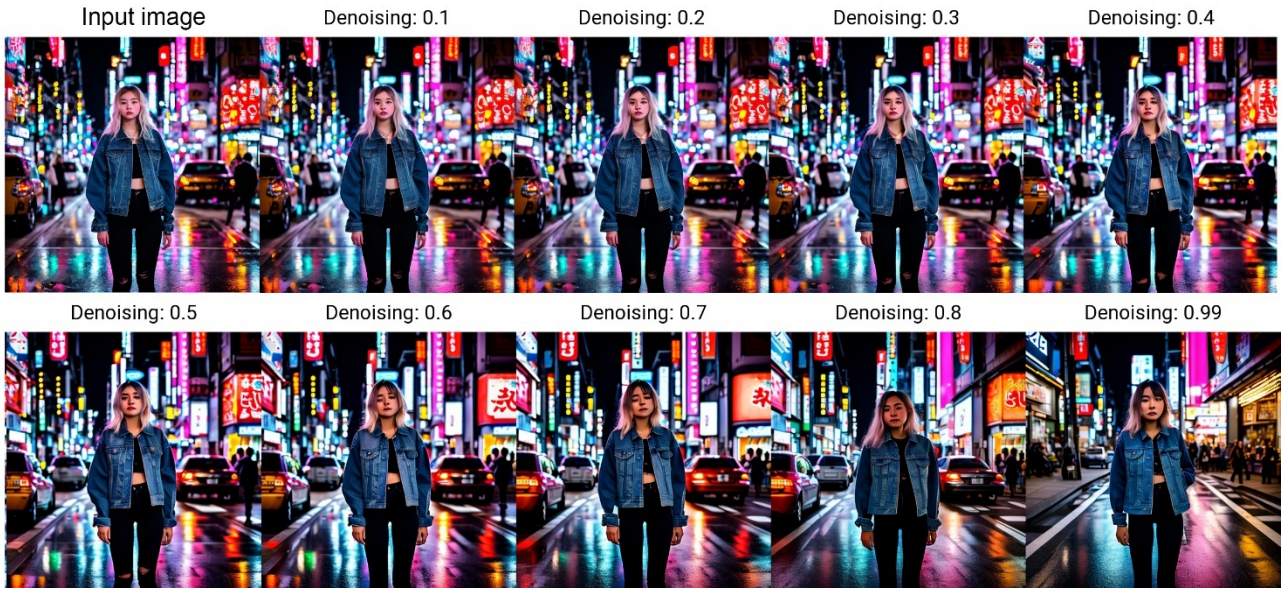

Рис. 3.18. Вплив різних значень denoising strength на вихідні зображення

Аналізуючи отримані приклади, можна зробити висновок, що використання різних значень параметра denoising strength призводять до створення варіацій вхідного зображення, де використання менших значень denoising strength (менше 0.5) дають зображення, що структурно більше схожі на вхідне зображення, проте значення вище (більше 0.5) призводять до більших змін у структурі вихідних зображень відносно вхідного зображення.

3.2.2. Створення зображень на основі текстового опису

Приклад створення зображень на основі текстового опису було наведено під час виконання пункту 3.2.1. Дослідження впливу параметрів генерації. Підсумуємо отримані у даному пункті результати для виконання поставленого завдання: використовуючи моделі латентної дифузії на основі текстового опису створити пару зображень «фотографія молодої дівчини, яка гуляє по вечірнім вулицям у Токіо. Дівчина повинна мати рожеве волосся середньої довжини,

синю джинсову куртку, чорні джинсові штани. Фон – типова вулиця японського мегаполісу, навколо дівчини є прохожі люди, вітрини та вивіски мають неонове освітлення» та «фотографія живописного пейзажу: зелений луг, за яким слідують високі гори, верхівки яких вкриті снігом. Час фотографії – вечір, червоно-рожеве небо на заході сонця». Параметри, що було використано для генерації наведених зображень наведено у таблиці 3.1.

Таблиця 3.1.

Параметри генерації для кожного із зображень завдання

Параметр	Перше зображення (Фотографія дівчини)	Друге зображення (фотографія пейзажу)
Prompt	A raw photo of a young lady, evening, in the middle of a street, Tokyo, street cold neon lights, denim jacket, black denim jeans, white shoes, public, people around, cold neon lighting,	A photo of a lanscape, (lush green valley:1.3), (green meadow:1.2) followed by a (snowy mountain:1.2), photorealistic, evening, sunset, pink and red sky, naturalism,
Negative prompt	[deformed | disfigured], poorly drawn, [bad : wrong] anatomy, [extra | missing | floating | disconnected] limb, (mutated hands and fingers), blurry, blurry face, shaded face. 3d. render,	3d, render, poorly drawn, low quality, worst quality, blurry
Steps	25	25
Sampler	DPM++ 2M Karras	DPM++ 2M Karras
CFG Scale	7	7.5
Seed	3931974505	2326489993
Size	512x512	768x512

Зображення до текстових описів із завдання, отримані при використанні кожного з наборів параметрів наведено на рис. 3.18.

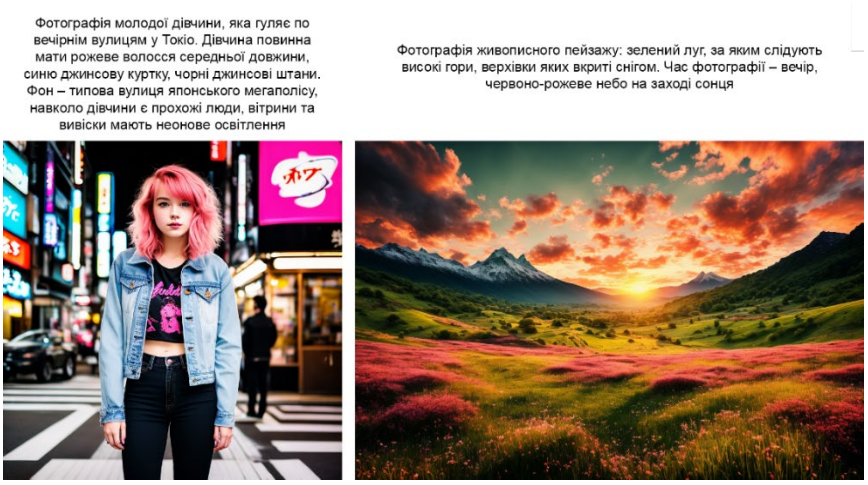

Рис. 3.18. Зображення до текстових описів, отримані при використанні кожного з наборів параметрів

Створення зображень на основі текстового опису є однією з ключових можливостей застосування моделей латентної дифузії, тому розуміння процесу налаштування параметрів генерації відіграє ключові роль у отриманні якісних вихідних зображень.

3.2.3. Створення зображень на основі вхідного зображення

Процес створення нових зображень на основі вхідних зображень (image-to-image) використовує текстовий опис та вхідне зображення. Процес має багато спільного із процесом створень зображень на основі текстового опису, додається лише новий вхід – вхідне зображення та один параметр – denoising strength, який визначає кількість випадкового шуму, що буде додано до представлення вхідного зображення під час ініціалізації процесу генерації.

Для створення зображень із вхідного зображення можна використати функцію image-to-image, яка генерує зображення на основі вхідного зображення та відповідного текстового опису. Даний процес генерації подібний до генерації зображення на основі текстового опису, додається лише новий вхід – зображення, яке потрібно використати та новий параметр – denoising strength,

який визначає скільки випадкового шуму буде додано до представлення вхідного зображення перед початком генерації.

Відтворимо отримане раніше зображення дівчини, використовуючи функцію image-to-image та модифікований текстовий опис. Замінимо локації та час доби у який було зроблено «знімки»:

--prompt "a raw photo of a lady, morning, in a park, green lush behind";

--prompt "a raw photo of a lady, in the dessert, mountains in the background,evening, peach sky";

--prompt "a raw photo of a lady, teacher, morning, daytime, in a school".

Таким чином, буде отримано три нових набори зображень, на яких повинно змінено навколишнє оточення та середовище: парк із зеленими гущами, пустеля із горами та школа. Приклад зображень, згенерованих на основі вхідного зображення наведено на рис. 3.20.

Рис. 3.20. Набори зображень згенеровані на основі вхідного зображення

Аналізуючи отримані результати, можна сказати, що функція image-to-image є корисним інструментом у контексті генеративного мистецтва, однак його використання сприяє значній відмінності між подібністю вхідного та вихідних зображень, тому використання даної функції не є рекомендованим, якщо стоїть мета більш точного структурного та композиційного відтворення вхідного

зображення. Далі ми розглянемо інші підходи до вирішення даної проблеми із використання моделей латентної дифузії.

3.2.4. Керування відтворенням зображення

Відтворення вхідного зображення представляє собою створення зображень, побідних до вхідного, наприклад, повторюючи композицію, розташування та пози об’єктів на вхідному зображенні. Даний підхід є корисним, якщо користувач вже має приклад структури та композиції зображень, які бажає отримати. Наразі є два основних способи відтворити вхідне зображення, використовуючи моделі латентної дифузії та їх розширення: використання розширень image-to-image або ControlNet.

3.2.4.2. Відтворення загальної композиції

Далі ми спробуємо відтворити згенероване раніше зображення дівчини, яка гуляє по вулицям Токіо. Вихідні зображення повинні зберегти позиціонування та позу дівчини у кадрі, але мати інший фон та зовнішній вигляд дівчини.

Найкращим способом для реалізації даної задачі є використання моделі ControlNet. ControlNet - це нейронна мережа, яка дозволяє користувачам керувати процесом генерації зображень з текстовими описами за допомогою Stable Diffusion. Вона працює шляхом додавання додаткового умовного входу до моделі Stable Diffusion, який можна використовувати для керування процесом генерації зображень різними способами. Таким чином, вхідними даними для дифузійної моделі є текстовий опис бажаного зображення, а для ControlNet – зображення, яке потрібно використати у якості базового прикладу композиції.

Для обробки вхідного зображення ControlNet використовує власні треновані нейронні моделі, які мають різний вплив на процес генерації зображень латентною моделлю дифузії. Найбільш вживаними є моделі

ControlNet Canny, Depth, NormalMap та OpenPose. Для кожної моделі існують власні пре-процесори, які виконують обробку вхідного зображення для перетворення його у представлення, яке зможе використати модель.

Приклад обробки вхідного зображення препроцесорами моделей Canny, Depth, NormalMap та OpenPose наведено на рис. 3.21.

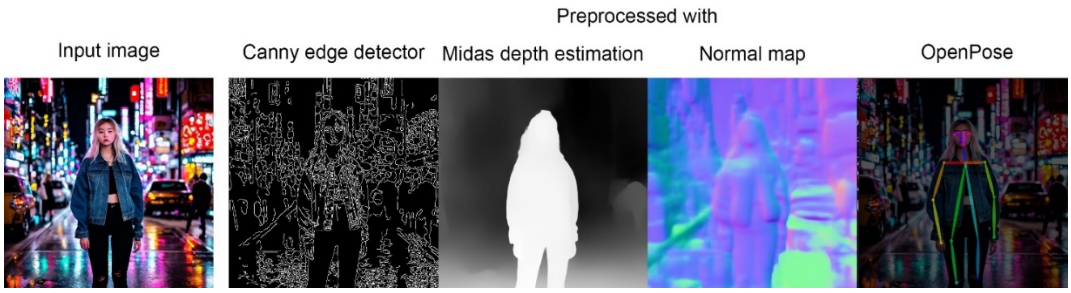

Рис. 3.21. Обробка вхідного зображення препроцесорами моделей Canny, Depth, NormalMap та OpenPose

Для контролю процесу генерації моделлю латентної дифузії, моделі ControlNet використовують наступні алгоритми:

– Canny базується на алгоритмі виявлення контурів (canny edge detection);

– Depth використовує алгоритм оцінки глибини (Midas depth estimation);

– NormalMap використовує алгоритм карт нормалей (normal maps);

– OpenPose використовує алгоритм OpenPose для виявлення поз об’єктів.

Маніпуляція над вхідними даними здійснюється модифікацією скелетного представлення пози002E

Створимо варіації вхідного зображення дівчини, базуючись на її положенні та позі у кадрі, використовуючи різні моделі ControlNet та варіанти текстових описів у запиті, приклад наведено на рис. 3.22.

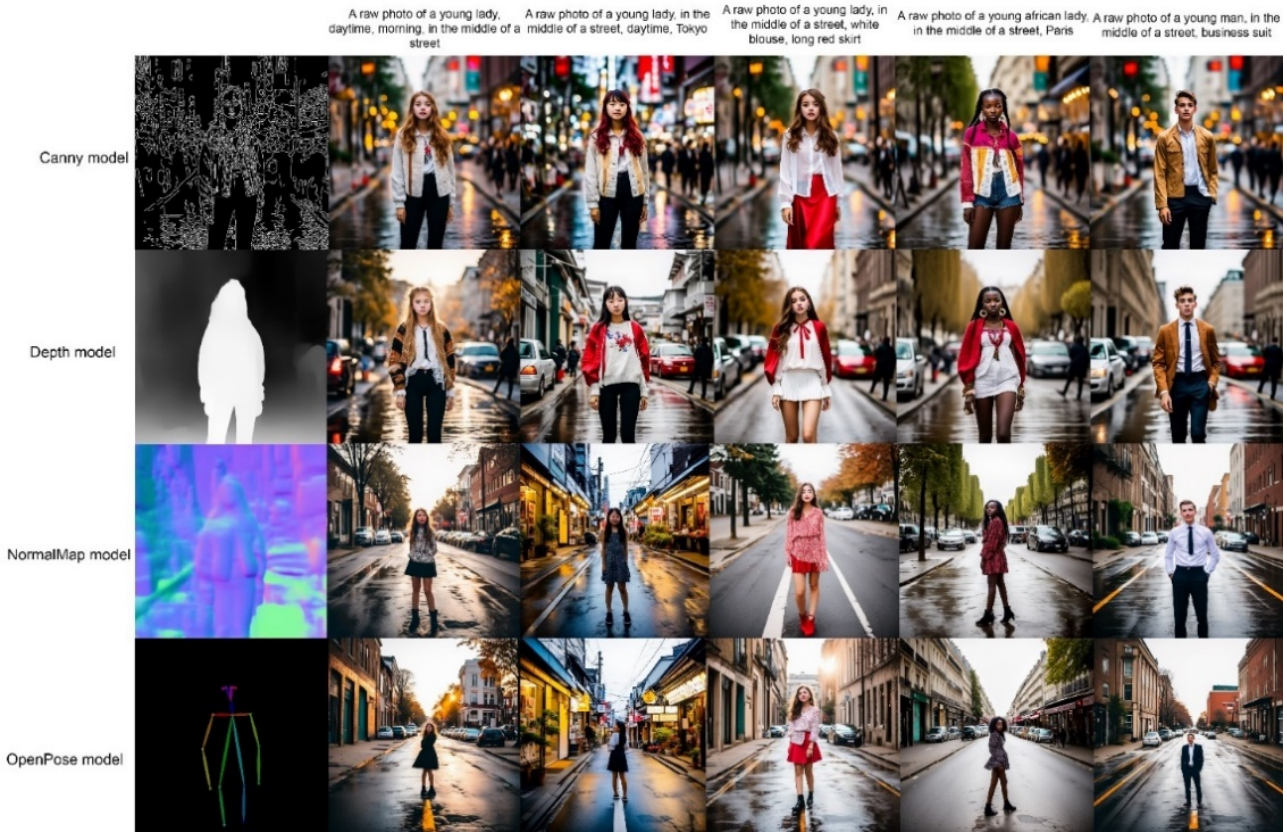

Рис. 3.22. Відтворення композиції: положення та пози дівчини із вхідного зображення

Можна побачити, що у даному випадку використання моделей Canny та Depth дали найкращі результати з модифікації вхідного зображення. Їх використання є оптимальним при відтворенні фотографій людей та їх розташування на зображенні.

3.2.4.3. Керування позами та положенням об’єктів

Для відтворення поз людей на зображенні буде використано модель OpenPose у складі ControlNet. Спочатку потрібно обрати бажану позу (придумати самостійно, або відтворити на основі існуючого зображення). Використання реальної фотографії є більш якісним джерелом, адже її повторення дозволить зберегти правильну людську анатомію на вихідних зображеннях.

Для додавання пози можна або завантажити готовий скелет пози та додати його до моделі OpenPose для керування процесом генерації, або створити

власний скелет. Для створення власного скелета слід відкрити OpenPose editor (розширення яке дозволяє створювати та модифікувати скелети-пози для роботи із OpenPose моделлю Controlnet).

Інтерфейс має такі налаштування, як розмір зображення, фонове зображення (щоб додати приклад та спростити відтворення пози), та контроль пози, який дозволяє створити новий скелет, та завантажити готовий. Процес створення скелету пози полягає у розташуванні вузлів скелета, кожен з яких відповідає певній кінцівці людського тіла, таким чином, щоб утворений скелет відповідав позі, яку бажає відтворити користувач. Приклад створення скелета-пози для моделі OpenPose із використання OpenPose Editor наведено на рис. 3.23.

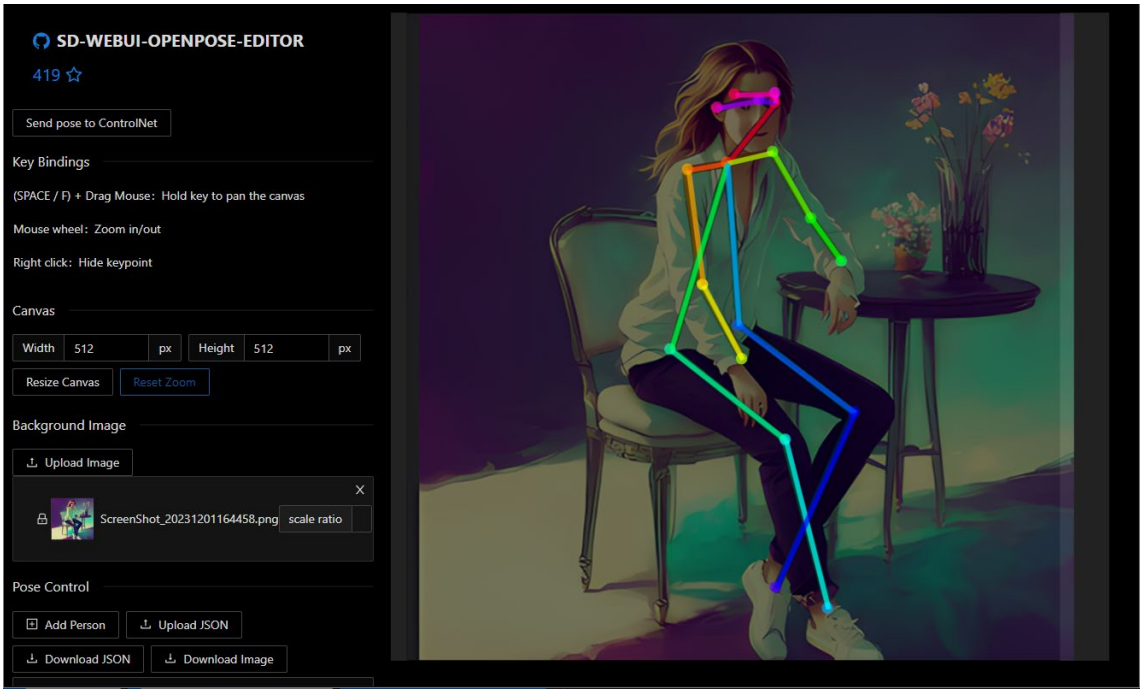

Рис. 3.23. Створення скелета-пози для моделі OpenPose із використання OpenPose Editor

Тепер відтворимо кілька поз, базуючись на зображеннях, знайдених у мережі Інтернет. Спочатку потрібно створити openpose скелет для кожної з бажаних до відтворення поз, після чого імпортувати його до OpenPose моделі ControlNet. Приклад джерел поз, створених поз-скелетів та вихідних зображень наведено на рис. 3.24.

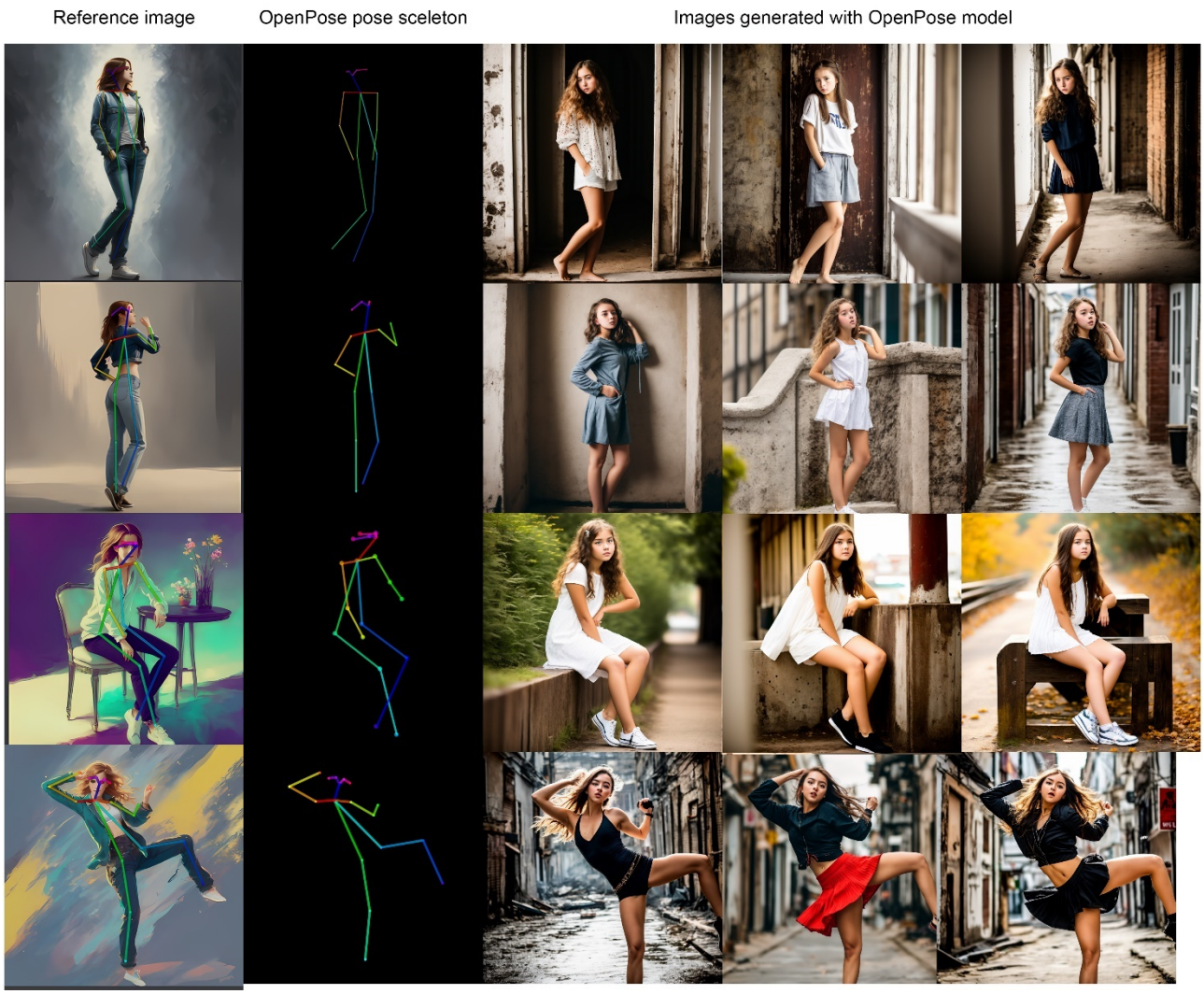

Рис. 3.24. Джерела оригінальних поз, їх пози-скелети та вихідні зображення, створені на їх основі, з використання моделі OpenPose

3.2.5. Пост-обробка згенерованих зображень

Пост-обробка згенерованих зображень це процес покращення отриманих під час генерації зображень із метою корекції наявних дефектів, покращення якості та збільшення роздільної здатності зображення.

3.2.5.1. Відновлення фрагментів зображення

Наскільки б якісною не була модель або текстовий опис, які використовує користувач, створення ідеального зображення з першої ж спроби є доволі складною задачею, оскільки моделі латентної дифузії є стохастичними, тобто

заснованими на випадковостях, створені зображення не завжди будуть відповідати тому, як їх собі уявляв користувач. Вихідні зображення можуть містити зайві деталі та елементи зображення, або їм навпаки не буде вистачати якихось важливих деталей, які користувач хотів би мати на зображенні.

Для вирішення даної проблеми є розширення для роботи з моделями латентної дифузії Inpaiting, яке дозволяє відновити фрагменти зображення, перемалювавши їх. Для його використання необхідного у вкладці image-to-image обрати Inpaint, після чого додати вхідне зображення, яке користувач бажає виправити. Для керування процесом відновлення/ перемальовування фрагментів зображення потрібно створити маску, використовуючи спеціальний пензлик. Області зображення, виділені у даній масці будуть змінені під час відновлення зображення, відповідно до встановлених налаштувань та текстового опису.

Inpaint приймає такі ж параметри як і text-to-image, однак додаються нові для налаштування роботи із маскою:

– mask blur – визначає попереднє розмиття виділеної область (у пікселях);

– mask mode – визначає яку область зображення потрібно відновити: ту, що виділено маскою, або навпаки;

– masked content – визначає чим буде заповнено виділену область перед її обробкою моделлю дифузії: оригінальне зображення або випадковий шум;

– only masked padding – визначає область зображення навколо маски у пікселях, яку буде прийнято до уваги моделлю дифузії під час обробки;

Видалимо оточуючих людей навколо дівчини на вхідному зображенні. Спочатку потрібно завантажити вхідне зображення, виділити область яку потрібно відновити, виставити параметри генерації (аналогічно до тих, що використовуються під час генерації у форматі image-to-image, але ще додаються наведені вище налаштування для роботи з маскою), написати текстовий отпис для фрагменту, виділеного маскою, у нашому випадку це було «tokyo street, road, building, street light». Після проведення налаштувань можна запустити обробку

зображення моделлю. Приклад використання функції Inpaint для відновлення фрагментів вхідного зображення та результат відновлення наведено на рис. 3.25.

Рис. 3.25. Використання функції Inpaint для відновлення фрагментів вхідного зображення та результат відновлення

Як можна побачити, функція Inpaint доволі непогано справляється із відновленням фрагментів зображення, створюючи контент, який вписується у загальну структуру та кольорові палітру вхідного зображення. Дана функція може бути використана для локальної корекції фрагментів зображення, наприклад, додаючи або прибираючи небажані елементи зображення.

3.2.5.2. Збільшення роздільної здатності та якості зображення

Вихідні зображення, згенеровані із малою роздільною здатністю (512x512 пікселів) можуть виглядати добре з точки зору їх структури та композиції, проте їх деталізація та якість не будуть задовільними. Для вирішення даної проблеми є певні підходи до пост-обробки згенерованих зображень із використанням моделей латентної дифузії та їх розширень із метою підвищення якості, деталізації та роздільної здатності вихідних зображень, наприклад Hires,fix або Ultimate Stable Diffusion Upscale.

3.2.5.2.1. Hires.Fix

Для збільшення роздільної здатності та якості зображень, згенерованих у режимі text-to-image є розширення Hires.Fix, яке дозволяє додатково обробити згенероване зображення із метою підвищення його якості, деталізації та роздільної здатності. Дане розширення приймає у якості вхідних даних згенероване зображення, після чого перемальовує його, використовуючи такі параметри налаштування як:

– hires steps – кількість кроків очищення шуму під час обробки вхідного зображення моделлю hires.fix;

– denoising strength – кількість доданого до вхідного зображення випадкового шуму (впливає на те наскільки зображення після обробки буде схоже на вхідне зображення);

– upscaler – модель, яку буде використано для збільшення роздільної здатності зображення. є різні моделі масштабування, кожна надає різну ступінь різкості вихідному зображенню; різні моделі можуть бути рекомендованими для різних типів зображень: фотореалістичні, ілюстрації тощо;

– resizing size – до якого розміру потрібно збільшити вхідне зображення. Задається як множник (scale), або значення висота та ширини;

Використаємо розширення Hires.fix для збільшення роздільної здпвтності вхідного зображення, встановивши наступні параметри: Hires steps – 25, denoising strength – 0.2, upscale by 2.

Приклад ретельного порівняння якості оригінального вихідного зображення та цього ж зображення після обробки моделлю hires.fix наведено на рис. 3.26.

Рис. 3.26. Порівняння якості та структури оригінального зображення та цього ж зображення після підвищення роздільної здатності у 1.5, 2 та 2.5 разів

Можна побачити, що якість та деталізацію вихідного зображення було значно покращено, до того ж було прибрано дефекти та спотворення у анатомії, особливо на обличчях. Однак, нажаль, зображення набувають певних змін, які роблять вхідне та вихідне зображення відмінними, До того ж, при спробі виконати збільшення розміру таким чином, щоб кожна зі сторін була більше ніж 1024 пікселі, на зображеннях можуть з’являтися небажані фрагменти, такі як дублювання об’єктів на зображенні. Тому, оптимальним є збільшення роздільної здатності зображення таким чином, шоб хоча б одна зі сторін була 1024 пікселі. Функція Hires,fix є корисним та ефективним інструментом для покращення якості та підвищення роздільної здатності згенерованих зображень.

3.2.5.2.2. Ultimate SD Upscale

Наступним методом покращення якості генерованого зображення є його обробка моделлю Ultimate SD Upscale, яке розбиває вхідне зображення на плитки заданої розмірності та поступово покращує їх якість, відповідно до заданих

параметрів та тестового опису. Дане розширення працює тільки у режимі image-to-image, та вимагає більше часу на обробку, адже вхідне зображення розбивається на сегменти, кожен з яких проходить повний цикл генерації: додавання випадкового шуму – прогнозування та видалення шуму – масштабування. Отже, чим більше вхідне зображення, тим більше часу потребує його обробка. Дане розширення приймає такі параметри як

– target size type – визначає розмір вихідного зображення;

– upscaler – визначає якою моделлю буде проведено масштабування;

– type – визначає порядок у якому буде виконуватися обробка сегнментів зображення (лінійно, або у шахматному порядку);

– tile width – розмір сегментів/плиток на які вхідне зображення буде розбито під час обчислень.

Для підвищення якості та роздільної здатності вхідного зображення спочатку виконаємо перший цикл покращення, після чого отримане зображення використаємо у якості нового вхідного зображення, та повторимо процес покращення. Використаємо наступні параметри:

– target size: scale from image size, scale = 2;

– Upscaler: 4x-UltraSharp

– Type: Linear;

– Tile width: 512 px;

– Padding: 32 px.

Приклад порівняння вхідного зображення та зображень, утворених після покращення вхідного зображення із використання Ultimate SD Upscale наведено на рис. 3.27.

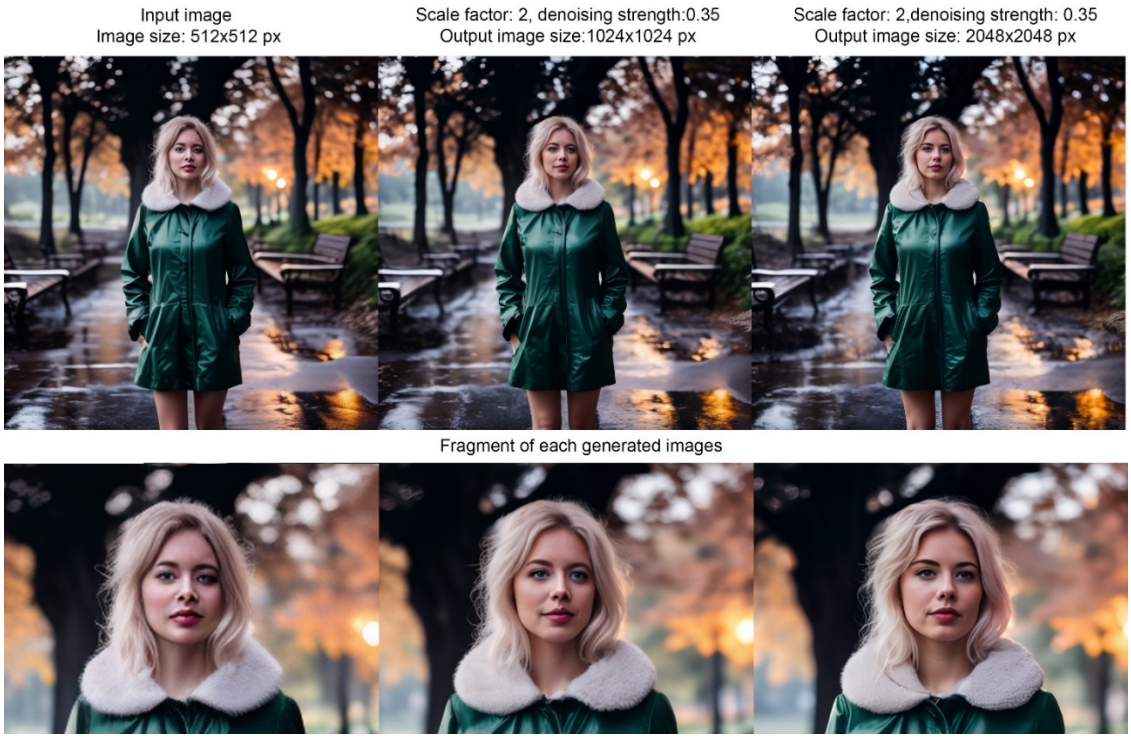

Рис. 3.27. Порівняння вхідного зображення та зображення, утвореного після використання Ultimate SD Upscale

Можна побачити, що після пост-обробки із використанням Ultimate SD upscale якість та деталізація вхідного зображення значно зросла, було виправлено певні помилки та спотворення у анатомії вхідного зображення, обличчя виглядають значно натуральніше, а отримані зображення мають бажану ступінь фотореалістичності. Використання розширення Ultimate SD upscale дозволяє значно покращити якість та деталізацію бажаного зображення, однак потребує значний витрат часу, адже під час обробки модель розбиває вхідне зображення на сітку зображень, кожне з яких проходить повний процес обробки, тому при збільшенні розміру зображення вдвічі, наприклад з 512x512 до 1024x1024 пікселів, буде створено 4 плитки, кожне з яких пройде повний цикл генерації. Дане розширення є ефективним інструментом для підвищення якості, деталізації та роздільної здатності зображення, однак через його значні витрати часу на обчислення при його використанні, слід використовувати його тільки коли користувач вже визначився із вхідним зображенням, яке бажає покращити, у іншому випадку використання даного розширення лише буде гальмувати

процес генерації зображень, адже краще спочатку створити велику кількість варіацій зображень, з яких вже можна обрати ті, що краще будуть відповідати побажанням користувача, замість того щоб намагатися покращити зображення, яке спочатку не відповідало потребам та побажанням.

3.3. Оптимізація параметрів генерації зображень при роботі з моделями латентної дифузії

Слід зазначити, що процес генерації зображень із використанням моделей латентної дифузії є доволі специфічним, оскільки алгоритм їх роботи передбачає використання випадкового початкового шуму та моделей, які виконують прогнозування та очищення шуму відповідно до того на якому наборі даних їх було треновано. Наведені чинники роблять неможливим попередній прогноз того як саме буде виглядати вихідне зображення. Через це неможливо сформувати універсальні рекомендації, які б дозволили користувачу завжди отримувати бажані зображення.

Процес підбору оптимальних параметрів генерації є ітеративним та базується на емпіричному досвіді користувача. Однак можна систематизувати отримані у практичному дослідженні результати таким чином, щоб надати користувачу вказівки, які дозволять пришвидшити етап підбору оптимальних параметрів генерації для створення якісних зображень та економії обчислювальник ресурсів. Вибір параметрів генерації залежить від потреб та побажань користувача до вихідних зображень, які він бажає отримати. Надалі буде сформовано рекомендації та вказівки налаштування основних параметрів генерації, що дозволять покращити якість вихідних зображень, пришвидшити етап підбору параметрів генерації та зменшити загальну кількість необхідної обчислювальної потужності за рахунок використання оптимальних параметрів генерації та зменшення загальної кількості необхідних ітерацій для визначення параметрів генерації, що дозволять користувачу згенерувати бажане зображення

при використанні моделей латентної дифузії у завданнях генеративного мистецтва.

3.3.1. Вибір моделі (Checkpoint)

Обрана попередньо тренована модель латентної дифузії є першим ключовим фактором, який впливатиме на увесь подальший процес генерації та вихідні зображення. Якщо обрану користувачем модель не було треновано на відповідному до потреб користувача наборі даних, обрана модель не зможе якісно (або, залежно від специфічності бажаного зображення – не зможе зовсім) згенерувати бажані зображення, які міститимуть бажані об’єкті або стилі.

Вибір моделі виконується відповідно до тематичних потреб, однак також приймати до уваги і базову модель, на основі якої було додатково треновано модель, яку користувач планує використовувати. Аналіз обраної моделі включає в себе:

– перевірку базової моделі, оскільки це визначає на якій розмірності вхідних зображень було треновано модель, та який розмір зображення буде оптимальним під час налаштування параметрів генерації;

– перевірку області застосування обраної моделі, щоб впевнитися, що обрана модель відповідає потребам користувача, адже моделі, які наприклад, були треновано для створення ілюстрацій та персонажів мультфільмів, не є рекомендованими для створення фотореалістичних зображень, та навпаки;

– слід ознайомитися із описом моделі та прикладами її використання, оскільки розробники часто залишають у описі інформацію стосовно процесу тренування моделі та рекомендації щодо налаштування процесу генерації, наприклад перелік слів із текстового опису, які найчастіше зустрічалися під час тренування моделі, та як наслідок, матимуть більший вплив на процес генерації та вихідне зображення. Також рекомендованим є ознайомитися з прикладами зображень, згенерованих із використанням даної моделі та параметрами генерації, що були використані.

Приклад перегляду інформаційної картки моделі на ресурсі Civitai.com наведено на рис. 3.28.

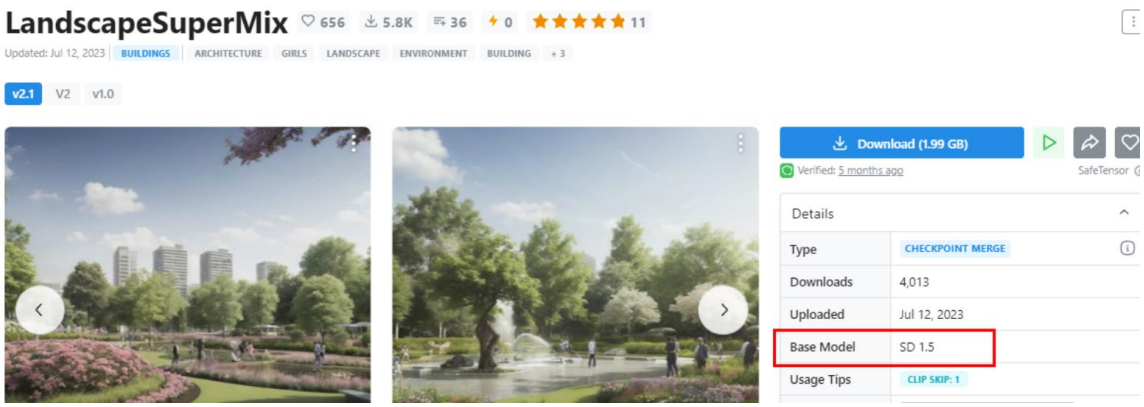

Рис. 3.28. Перегляд специфікації моделі та її базової моделі

Основну інформацію про моделі Stable Diffusion, включаючи назву тренувального набору даних, розмірність зображень під час тренування, кількість кроків тренування на кожному з наборів зображень та базова версія-предок, на базі якої було треновано дану модель наведено у таблиці 3.2.

Таблиця 3.2.

Моделі Stable Diffusion [45, 46, 47]

Model	Training Dataset	Training resolution	Steps	Resume of
SD v1.1	Laion2B-en Laion-high-resolution	256x256 512x512	237 000 194 000	-
SD v1.2	Laion-improved-aesthetics	512x512	515 000	v-1-1
SD v1.3	laion-improved-aesthetics	512x512	195 000	v-1-2
SD v1.4	laion-aesthetics v2-5+	512x512	225 000	v-1-2
SD v1.5	laion-aesthetics v2-5+	512x512	595 000	v-1-1
SDv2.0base	LAION-5B	256x256 512x512	550 000 850 000	-
SD v2.0	LAION-5B	768x768		v2.0 base
SDv2.1base	LAION-5B	512x512		V2.0
SD v2.1-v	LAION-5B	768x768	210 000	V2.0
SDXL0.9base	LAION-5B	1024x1024		-
SDXL1.0base	LAION-5B	1024x1024		SDXL0.9base

3.3.2. Текстовий опис (Prompt)

Текстовий опис повинен детально, проте лаконічно відображати зображення, яке користувач бажає отримати. Текстовий опис складається із слів-токенів, розділених комою. Кожен токен із текстового опису буде співвідноситися із текстовим описом зображень, із набору даних на якому було треновано модель. Слід зазначити, що порядок слів-токенів у текстовому описі впливає на те, скільки модель буде приділяти їм уваги під час процесу генерації. Таким чином, під час написання текстового опису слід враховувати слова, якими виконується опис та їх положення у описі. Найбільший пріоритет мають токени, що знаходяться на початку текстового опису (ліворуч), та по мірі їх віддалення від початку їх вплив зменшується. Однак, токени, які знаходяться ближче до кінця опису, впливають на усі токени, що були поперед них, тому, зазвичай, у першу чергу у текстовому описі слід вказати ключові об’єкти бажаного зображення та їх характеристики, а лише потім виконуються опис навколишнього середовища, стилістики, освітлення зображення, тощо. Загальна структура якісного текстового опису виглядає наступним чином:

– subject – визначити об’єкти, які користувач хоче бачити на зображенні;

– medium – конкретизувати опису об’єкта, наприклад, для людини – опис предметів одягу, зовнішнього вигляду, віку, положення, її дії тощо;

– style – визначити у якому стилі має бути вихідне зображення, наприклад фотореалізм, картина фарбою, скульптура з каменю, тощо. також додатково можна вказати художника, стиль якого користувач бажає відтворити (за умов що моделі бачила відповідні прикладу під час тренування);

– additional details – можна додатково описати бажані деталі, що повинні бути присутні на зображенні

– resolution – дозволяє вплинути на якісні характеристики вихідного зображення, наприклад, додавши highly detailed, sharp focus, best quality, best anatomy, masterpiece можна значно покращити якість вихідного зображення;

– lighting – визначити яке освітлення потрібно додати до зображення, час доби, джерело та напрямок світла, стиль освітлення, наприклад кінематографічний.

3.3.3. Запобігання небажаним зображенням (Negative prompt)

Negative prompt запобігає появі на зображенні небажаних об’єктів та деталей, які не потрібно додавати. Вплив порядку токенів у текстовому описі Negative prompt є аналогічним до правил, якими керується Prompt: токени ближче до початку мають більшу вагу та пріоритет, та навпаки. Написання текстового опису Negative prompt повинно відповідати тематиці та вимогам до бажаного зображення, враховуючи особливості зображення та його об’єктів. Наприклад, для створення фотографій людей та фотографій ландшафту, потрібні різні текстові описи Negative prompt, оскільки об’єкти на зображеннях мають різну структуру: для зображення людини бажано включити до Negative prompt опис того, як анатомію людини не має бути спотворено - запобігти появі зайвих кінцівок, ненатуральній анатомії, ненатуральним позам тощо, однак для зображення ландшафту опис людини не буде мати ніякої користі.

Таким чином, ключем до написання якісного Negative prompt є досвід користувача, послідовний та структурний підхід до формування текстового опису відповідно до семантичної складової зображення, та аналіз результатів, отриманих при використанні різних варіантів текстового опису Negative prompt.

3.3.4. Керування відповідністю текстовому опису (CFG Scale)

Значення параметра CFG Scale для кожного окремого випадку може визначатися ітеративно, шляхом створення зображень із різними значеннями даного параметра, для визначення оптимального значення, яке буде надавати вихідні зображення, які відповідають потребам користувача. Однак,

оптимальним є використання значень у діапазоні від 7 до 10. Стисло про кожен з можливих діапазонів значень параметра CFG:

– діапазон значень [1 .. 6] забезпечує створення моделлю більш креативних зображень із більшою варіативністю відносно текстового опису, але вихідні зображення можуть мати структурні та анатомічні спотворення, бліду ненатуральну кольорову палітру та не відповідатимуть текстовому опису. Використання значень із даного діапазону не є рекомендованим;

– діапазон значень [7 .. 10] є рекомендованим для більшості запитів. Значення у даному діапазоні дозволяють зберегти оптимальний баланс між креативністю моделі під час генерації та відповідністю текстовому опису, забезпечуючи створення якісних та структурно правильних зображень;

– діапазон значень [11 .. 15] є менш рекомендованим, винятком є випадки, коли текстовий опис має достатню деталізацію бажаного зображення і дуже чітко вказує, як має виглядати вихідне зображення, у інших випадках слід відмовитися від використання значень даного діапазону, оскільки це може призводити до появи спотворень, які роблять вихідні зображення ненатуральними, наприклад через занадто високу контрастність;

– діапазон значень [16 .. 20] є ще більш чутливим до детальності та якості текстового опису, тому використання значень параметра CFG із даного діапазону загалом не рекомендується, оскільки може негативно впливати на якість вихідного зображення, призводячи до появи спотворень та артефактів на вихідному зображенні;

– діапазон значень [20 .. 30] майже ніколи не придатні, через те, їх використання призводить до значного спотворення та появи артефактів на вихідних зображеннях.

3.3.5. Кількість кроків очищення шуму (Sampling steps)

Кількість кроків обробки визначає кількість ітерації прогнозування та очищення шуму із вхідного представлення зображення. Мала кількість кроків не

дозволить алгоритму очищення шуму видалити достатньо шуму для утворення зображення бажаної якості та деталізації. Занадто велика кількість кроків очищення шуму призведе до появи зайвих елементів та деталей на вихідному зображенні, та суттєво збільшить витрати часу на генерацію зображень.

Можна виділити такі основні діапазони можливих значень:

– значення у діапазоні [1 .. 10] будуть недостатніми для видалення достатньо кількості шуму із латентного представлення зображення. Вихідні зображення не будуть придатними до використання через занадто значну кількість деформації, спотворень та графічних артефактів. Використання значень з даного діапазону категорично не рекомендується;

– значення у діапазоні [11 .. 19] створюють вихідні зображення задовільної якості, проте вони будуть містити певну кількість розмитих елементів зображення та артефактів. Однак, за рахунок менших витрат часу на обчислення, значення з даного діапазону можуть бути використані щоб оцінити якість текстового опису та приблизно спрогнозувати як буде виглядати більш якісне зображення, згенероване при більшій кількості кроків очищення. Значення з даного діапазону можуть бути використані, якщо немає високих вимог до якості вихідних зображень, або для створення проміжних варіантів бажаного зображення під час ітеративного визначення оптимального текстового опису та решти параметрів генерації;

– значення у діапазоні [20 .. 35] дають найкращі вихідні результати, особливо у співвідношенні часу, витраченого на генерацію, до якості вихідного зображення. Даний діапазон значень є оптимальним і рекомендованим до застосування під час процесу генерації зображень;

– значення у діапазоні [36 .. 100] можуть давати вихідні зображення із різною якістю та деталізацією, які не так сильно піддаються спотворенню. Проте використання такої кількості очищення шуму потребує більше часу, та майже не впливає на якість вихідних зображень. Слід уникати значень із даного діапазону, окрім випадків коли під час генерації зображень із меншою кількості кроків очищення вихідні зображення зберігають спотворення або недостатню якість.

3.3.6. Алгоритми очищення шуму (Sampling methods)

Вибір алгоритму очищення шуму впливатиме як на якість зображення, так і на час виконання кроків очищення шуму, що впливатиме на загальну кількість часу, необхідного на виконання генерації. Немає «правильного» алгоритму очищення шуму, різні алгоритми можуть трохи змінювати вихідне зображення, тому оптимальним буде ітеративно перевірити результати, які надає використання кожного з алгоритмів очищення та обрати той, що буде відповідати вимогам користувача. Однак, можна виділити найуживаніші алгоритми очищення шуму, які використовуються найчастіше серед розробників та ентузіастів у сфері генеративного мистецтва, які працюють із моделями латентної дифузії, та відмічають якість вихідних зображень. Тому, у якості початкового вибору слід зупинитися на наступних алгоритмах: Euler, Euler a, алгоритми групи DPM++, такі як DPM ++ 2M Karras та DPM++ SDE Karras. Приклад зображень, згенерованих із застосуванням різних алгоритмів очищення шуму при різній кількості кроків обробки наведено на рис. 3.29.

Рис. 3.29. Зображення, згенеровані із застосуванням різних алгоритмів очищення шуму: Euler, Euler a, DPM ++ 2M Karras та DPM++ SDE Karras

Детальніше про ключові алгоритми очищення шуму:

– Euler – дає якісні результати, які, менш схильні до появи розбіжностей від вихідними зображеннями при повторній генерації, потребує менше часу на

виконання обчислень відносно алгоритмів групи DPM. Даний алгоритм є рекомендованим для створення фотореалістичних зображень;

– Euler a – дає якісні результати, проте більше схильний до появи розбіжності серед вихідних зображень при повторній генерації через додавання нового випадкового шуму на кожному кроці очищення, потребує менше часу на виконання обчислень відносно алгоритмів групи DPM. Даний алгоритм є рекомендованим для створення фотореалістичних зображень;

– Алгоритми групи DPM++, такі, як, наприклад, DPM++ SDE Karras та DPM++ 2M створюють якісні зображення навіть при меншій кількості кроків, однак потребують більше часу на обчислення та можуть мати певну розбіжність у вихідних зображеннях згенерованих при тих же значеннях параметрів генерації Дані алгоритми є рекомендованими для створення ілюстрацій та малюнків.

Отже, у якості початкового алгоритму очищення шуму користувачу слід обирати один із наступних алгоритмів: Euler, Euler a, алгоритми групи DPM++, такі як DPM ++ 2M Karras та DPM++ SDE Karras, віддаючи перевагу алгоритмам очищення групи Euler при створенні фотореалістичних зображень або якщо є потреба у скороченні часу витраченого на генерацію зображення, та використовувати алгоритми групи DPM++ у ситуаціях коли виконується генерація ілюстрацій та малюнків, особливо при використанні кількості кроків обробки менше за рекомендовану (20-35 кроків).

3.3.7. Розмір зображення (Image size)

Оптимальним є використання значень розміру зображення, які відповідають розмірності зображень на яких було треновано обрану для використання під час генерації модель латентної дифузії. Розмірність зображень, що було використано під час тренування базових моделей латентної дифузії можна переглянути у таблиці 3.2.

Наразі, більшість існуючих моделей латентної дифузії, було додатково треновано на базі моделі Stable Diffusion v1.5, тому оптимальним є встановлення

значення розміру генерованого зображення 512x512 пікселів, або таким чином, щоб хоча б одна зі сторін дорівнювала 512 пікселям. Використання інших розмірів може негативно вплинути на якість та структуру вихідних зображень, що призведе до втрати якості, деталізації або дублювання об’єктів на вихідних зображеннях.

Для генерації зображень із більшою роздільною здатністю правильним підходом буде генерація зображення, де хоча б одна зі сторін буде відповідати розмірності зображень, використаних у наборі даних під час навчання моделі, та подальше пост-обробка вихідного зображення за алгоритмами, описаними у пункті 3.2.5. Пост-обробка згенерованих зображень.

3.3.8. Значення початкового шуму (Seed)

Значення даного параметра не є обов’язковим до завдання вручну, якщо стоїть цілі створити якомога більше варіантів зображень на основі наявного текстового опису. Використання того ж значення seed при ідентичних налаштування решти параметрів генерації дозволяє відтворити зображення, згенероване іншими користувачами. У випадку, коли конкретне вихідне згенероване зображення відповідає вимогам користувача, однак, він все ж бажає внести певні зміни до структури або деталізації зображення, слід використати значення seed, ідентичне тому що було використано під час генерації оригінального зображення, щоб забезпечити відтворюваність зображення.

Після налаштування значення параметра seed можна поступово змінювати текстовий опис та інші параметри генерації (окрім розміру та моделі), якими керується модель під час генерації зображення. Це дозволить вносити зміни до вихідного зображення, зберігаючи його оригінальну структуру та ітеративно покращуючи якість, деталізацію та відповідність побажанням та вимогам користувача.

Таким чином, значення seed слід залишати випадковим у ситуаціях коли користувачу потрібно створити варіації вихідних зображень до наданого

текстового опису, та навпаки, встановлювати фіксоване значення seed, якщо стоїть задача відтворення та модифікації вихідних зображень.

3.3.9. Значення доданого шуму (Denoising strength)

Значення доданого шуму визначає скільки нового випадкового шуму буде додано до представлення вхідного зображення, наприклад при генерації у режимі Image-To-Image, або під час використання функції Hires,fix.

Можна сформувати наступні групи можливих діапазонів значень даного параметра та їх впливу на процес генерації та вихідне зображення:

– діапазон значень [0.01 .. 0.19] призводить до того, що вихідні зображення будуть мати мінімальні відмінності відносно вхідного зображення, однак, це не дає моделі змоги виправити наявні дефективні деталі або додати нові деталі до зображення, тому значення з даного діапазоні є менш рекомендованими до використання, окрім випадків коли вхідне зображення не містить спотворень та потребує значної модифікації;

– діапазон значень [0.2 .. 0.49] дозволяє внести певні модифікації до зображення, при цьому зберігаючи його загальну структуру та композицію. Є рекомендованими, якщо структура вхідного зображення влаштовує користувача, однак він хотів би внести певні зміни до якості або деталізації зображення;

– діапазон значень [0.50 .. 0.75] призводить до створення зображень, що значно відрізняються від вхідного зображення. Можуть бути використані для створення варіацій вхідного зображення, які будуть мати схожу композицію та кольорову палітру, однак будуть містити суттєві структурні відмінності;

– діапазон значень [0.76 .. 0.99] призводить до створення зображень, які майже не будуть мати спільних рис із вхідним зображенням. Використання даних значень може бути обґрунтовано специфічними потребами користувача, але у загальному випадку їх використання не є рекомендованим через занадто велику варіативність відносно вхідного зображення, що критично змінює структуру оригінального зображення. У більшості випадків замість

використання даного діапазону значень більш ефективним буде використання функції Text-To-Image для створення нових зображень, які будуть відповідати текстовому опису та вимогам користувача.

3.4. Висновок до третього розділу

У третьому розділі було практично досліджено процес генерації зображень із використанням моделей латентної дифузії з метою дослідження впливу параметрів генерації на вихідне зображення. Додатково було розглянуто можливі способи практичного застосування сервісів, які використовують моделі латентної дифузії для генерації зображень.

У якості практичного використання моделей латентної дифузії було проведено генерацію зображень відповідно до поставленого завдання: генерація зображень на основі текстового опису, генерація зображень на основі зображень, відтворення вхідних зображень, відтворення композиції вхідного зображення та відтворення поз та розташування об’єктів на вхідних зображеннях, та пост-обробка згенерованих зображень із метою підвищення їх якості та роздільної здатності.

Під час виконання практичного дослідження процесу генерації зображень було досліджено основні параметри генерації, такі як текстовий опис, негативний опис, використання різних моделей, керування відповідністю текстовому опису, кількість кроків очищення шуму та алгоритми очищення шуму, розмірність зображення та значення початкового шуму. Було досліджено призначення параметрів генерації та вплив налаштування їх значень на процес генерації та вихідні зображення. метою визначення їх впливу

Отримані під час практичного дослідження результати було проаналізовано, та на основі отриманих даних було розроблено перелік рекомендацій для налаштування параметрів генерації, які дозволили б забезпечити створення якісних зображень та пришвидшити процес генерації із

зменшенням необхідної обчислювальної потужності із використанням моделей латентної дифузії.

Сформовані у даному розділі рекомендації та вказівки можуть бути практично використані під час процесу генерації зображень при використанні різних версій та модифікацій моделей латентної дифузії із метою покращення якості вихідних зображень, оптимізації етапу підбору оптимальних параметрів та зменшення кількості необхідних обчислювальних ресурсів у задачах генеративного мистецтва, відповідно до цілей та вимог користувача. Дані рекомендації дозволять користувачу отримати систематизовані теоретичні та практичні знання та навички, які необхідні при роботі із моделями латентної дифузії для оптимізації та пришвидшення процесу створення зображень у задачах генеративного мистецтва.

Використання сформованих рекомендацій дозволить користувачам ефективно керувати налаштування параметрів генерації та процесом генерації для створення якісних вихідних зображень, дозволивши при цьому зменшити час та кількість ітерацій, витрачених на процес створення бажаного зображення, при цьому значно скоротивши витрати необхідних для генерації зображень обчислювальних ресурсів. Зменшення кількості необхідних обчислювальних ресурсів відбувається за рахунок оптимізації етапу підбору значень параметрів генерації, адже сформовані рекомендації та вказівки дозволяють користувачу заздалегідь обрати значення кожного із параметрів генерації із його оптимального діапазону значень, що у свою чергу робить робочий процес більш зрозумілим та послідовним, значно скорочуючи кількість ітерацій генерації зображень, необхідних для визначення параметрів, що дозволять отримати зображення, які влаштовували би користувача.

ВИСНОВКИ

Метою кваліфікаційної роботи є теоретичне та практичне дослідження моделей латентної дифузії у контексті генеративного мистецтва із метою глибокого аналізу та розуміння технічних аспектів роботи даного класу генеративних моделей та формування переліку рекомендацій для налаштування параметрів, які використовуються моделлю латентної дифузії під час генерації зображень із метою підвищення їх якості та швидкості генерації.

У ході виконання даної кваліфікаційної роботи було проведено аналіз популярних підходів до процесу генерації зображень, результатом чого став висновок про зростаючу популярність та широке використання моделей, заснованих на дифузії. Але дані моделі є занадто вибагливими до технічних показників обчислювальної машини на якій вони використовуються. Враховуючи це, використання моделей латентної дифузії, досліджених у даній кваліфікаційній роботі, є більш оптимальним рішенням через свою інноваційну архітектуру, яка дозволила значно скоротити витрати на навчання даного класу моделей та знизити вимоги до технічних характеристик обчислювальних машин на яких вони працюють. На основі проведеного дослідження можна зробити наступні висновки:

– моделі латентної дифузії є одним із найперспективніших підходів до генерації зображень, оскільки вони дозволяють створювати зображення високої якості, а їх використання є ефективним з точки зору необхідних обчислювальних ресурсів, відносно інших генеративних моделей;

– моделі латентної дифузії можуть бути використані для генерації різних типів зображень, включаючи реалістичні або абстракті зображення;

– значення встановлених параметрів генерації мають ключовий вплив на якість вихідних зображень та швидкість їх створення. Через це етап налаштування параметрів генерації може бути складним та незрозумілим для користувача, який не має достатньо практичного досвіду у роботі із генеративними моделями, зокрема моделями латентної дифузії.

Результатом виконання даної кваліфікаційної роботи є сформований на основі практичних та теоретичних досліджень перелік рекомендації з налаштування параметрів генерації, які використовуються під час генерації зображень із використанням моделей латентної дифузії. Розроблені рекомендації можуть бути практично використані при роботі з будь-якими моделями латентної дифузії у контексті генеративного мистецтва, з метою підвищення якості вихідних зображень, оптимізації процесу підбору значень параметрів генерації, та як наслідок, зменшення необхідної кількості обчислювальних ресурсів, забезпечуючи створення якісних зображень, які відповідають потребам та вимогам потенційного користувача.

Результати теоретичного дослідження можуть бути використані для покращення користувачем розуміння природи та внутрішніх алгоритмів роботи моделей латентної дифузії, які застосовуються під час навчання або використання даного класу генеративної моделі. Сформовані у результаті практичного дослідження рекомендації та вказівки можуть бути використані для підвищення якості генерованих зображень при роботі із моделями латентної дифузії. Використання рекомендацій, розроблених у рамках даного дослідження, дозволяє оптимізувати процес генерації зображень із використанням моделей латентної дифузії, забезпечивши створення якісних вихідних зображень, зменшення часу та кількості ітерацій, витрачених на процес створення бажаного зображення, при цьому значно скоротивши витрати необхідних для створення зображень обчислювальних ресурсів.

Таким чином, використання сформованих рекомендацій дозволяє користувачу уникнути ситуацій, коли згенеровані зображення через неоптимальні значення параметрів генерації, не будуть відповідати вимогам до якості, деталізації, структури, відповідності текстовому опису, тощо. Адже використання значень параметрів, що менші значень із рекомендованого діапазону призведе до створення зображень низької якості, які будуть містити значну кількість спотворень. Та навпаки, використання значень параметрів генерації, які вище за рекомендований діапазон значень, будуть призводити до

спотворень та збільшення витрат часу та обчислювальних ресурсів, витрачених на створення набору зображень.

Таким чином, використання значень параметрів генерації із неоптимального діапазону призводить до створення зображень, які не відповідають вимогам до якості та деталізації, та надлишкових витрат часу та обчислювальних ресурсів, оскільки кожна ітерація генерації зображення, яка призвела до небажаних результатів, які не можуть бути використані, вимагає певних витрат обчислювальної потужності. Без використання систематизованих рекомендацій та вказівок стосовно налаштування параметрів генерації, процес генерації зображень втрачає ефективність через значну кількість ітерацій генерації, яку користувачу слід виконати перш ніж він нарешті отримає бажані зображення, які відповідатимуть вимогам якості, деталізації, стилізації та відповідності текстовому опису.

Перспективні напрямки досліджень у галузі моделей латентної дифузії:

– розробка нових методів навчання моделей латентної дифузії, які дозволили б генерувати зображення ще вищої якості;

– розширення можливостей моделей латентної дифузії, наприклад, для генерації зображень із заданими властивостями або для створення зображень, які відповідають певному стилю;

– дослідження впливу моделей латентної дифузії на розвиток генеративного мистецтва.

Таким чином, моделі латентної дифузії є потужним та перспективним інструментом для вирішення задач генеративного мистецтва, дозволяючи створювати фотореалістичні зображення та ілюстрації високої якості, які схожі на ті, що були створені людиною.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning – 2016 – P. 98.

2. Generative Adversarial Networks: A Brief History and Overview. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/370357461_Generative_Adversarial_Networks_A_Brief_History_and_Overview

3. Tero Karras, Samuli Laine, Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks – 2019. [Електронний ресурс]. – Режим доступу : https://www.researchgate.net/publication/338514531_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks

4. Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. Image Style Transfer Using Convolutional Neural Networks. 2016. [Електронний ресурс]. – Режим доступу: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf

5. Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. 2020. [Електронний ресурс]. – Режим доступу: https://ui.adsabs.harvard.edu/abs/2020arXiv200206177M/abstract

6. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks. 2014. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1406.2661

7. Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, Timo Aila. Training Generative Adversarial Networks with Limited Data. 2020. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2006.06676

8. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1706.03762

9. Diederik P. Kingma, Max Welling. Auto-Encoding Variational Bayes. 2013. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1312.6114.

10. Офіційна документація Stable Diffusion. [Електронний ресурс]. – Режим доступу: https://huggingface.co/blog/stable_diffusion

11. Офіційна документація MidJourney. [Електронний ресурс]. – Режим доступу: https://docs.midjourney.com/docs/quick-start

12. Офіційна документація Dall-E 2. [Електронний ресурс]. – Режим доступу: https://help.openai.com/en/collections/3557252-dall-e

13. Офіційна документація Google DeepDream. [Електронний ресурс]. – Режим доступу: https://blog.research.google/2015/06/inceptionism-going-deeper-into-neural.html

14. Офіційна документація Nvidia GauGAN. [Електронний ресурс]. – Режим доступу: https://blogs.nvidia.com/blog/2022/03/01/what-is-gaugan-ai-art-demo/

15. Lilian Weng. What are Diffusion Models? - 2021. [Електронний ресурс]. – Режим доступу : https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

16. Jonathan Ho, Ajay Jain, Pieter Abbeel. Denoising Diffusion Probabilistic Models - 2020. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2006.11239

17. CompVis. Офіційна документація моделей stable-diffusion на платформі GitHub. [Електронний ресурс]. – Режим доступу: https://github.com/CompVis/stable-diffusion

18. Akruti Acharya. An Introduction to Diffusion Models for Machine Learning – 2023. [Електронний ресурс]. – Режим доступу: https://encord.com/blog/diffusion-models/

19. Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models - 2021. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2112.10752

20. Onkar Mishra. Stable Diffusion Explained – 2023. [Електронний ресурс]. – Режим доступу: https://medium.com/@onkarmishra/stable-diffusion-explained-1f101284484d

21. Edmond Yip. What Is VAE in Stable Diffusion? – 2023. [Електронний ресурс]. – Режим доступу: https://builtin.com/artificial-intelligence/stable-diffusion-vae

22 Diederik P. Kingma, Max Welling. An Introduction to Variational Autoencoders – 2019. [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1906.02691.pdf

23. Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation – 2015. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1505.04597

24. Sairam Sundaresan. You Can’t Spell Diffusion without U – 2022. [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/you-cant-spell-diffusion-without-u-60635f569579

25. Aleksa Nikolić. What Is Stable Diffusion and How Does It Work? – 2023. [Електронний ресурс]. – Режим доступу: https://www.vegaitglobal.com/media-center/knowledge-base/what-is-stable-diffusion-and-how-does-it-work

26. Yuxuan Ding, Chunna Tian, Haoxuan Ding, Lingqiao Liu. The CLIP Model is Secretly an Image-to-Prompt Converter – 2023. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2305.12716

27. Maximilian Schreiner. New CLIP model aims to make Stable Diffusion even better – 2022. [Електронний ресурс]. – Режим доступу: https://the-decoder.com/new-clip-model-aims-to-make-stable-diffusion-even-better/

28. Sergios Karagiannakos,Nikolas Adaloglou. How diffusion models work: the math from scratch – 2022. [Електронний ресурс]. – Режим доступу: https://theaisummer.com/diffusion-models/

29. Hadar Sharvit. Latent Diffusion Models - 2023. [Електронний ресурс]. – Режим доступу: https://www.itshadar.com/latent-diffusion-models/

30. Sertis. Latent Diffusion Models: A Review — 2023. [Електронний ресурс]. – Режим доступу: https://sertiscorp.medium.com/latent-diffusion-models-a-review-part-i-d0feacc4906

31. Jay Alammar. The Illustrated Stable Diffusion – 2022. [Електронний ресурс]. – Режим доступу: https://jalammar.github.io/illustrated-stable-diffusion/?ref=gptechblog.com

32. Офіційна документація від HuggingFace. Text-to-image. [Електронний ресурс]. – Режим доступу: https://huggingface.co/docs /diffusers/using-diffusers/conditional_image_generation

33. OpenArt blog. The Most Complete Guide to Stable Diffusion Parameters – 2022. [Електронний ресурс]. – Режим доступу: https://blog.openart.ai/ 2023/02/13/the-most-complete-guide-to-stable-diffusion-parameters/

34. StableDiffusionArt blog. Know these Important Parameters for stunning AI images – 2023. [Електронний ресурс]. – Режим доступу: https://stable-diffusion-art.com/know-these-important-parameters-for-stunning-ai-images/

35. Novita.Ai. Train Prompts Stable Diffusion: A Comprehensive Guide – 2023. [Електронний ресурс]. – Режим доступу: https://medium.com/@novita_ai/ train-prompts-stable-diffusion-a-comprehensive-guide-758b906fcbd9

36. Novita.Ai. Stable Diffusion Seed: The Ultimate Guide – 2023. [Електронний ресурс]. – Режим доступу: https://medium.com/@novita_ai/stable-diffusion-seed-the-ultimate-guide-a7ada09a932d

37. Jon Martindale. Stable Diffusion PC system requirements – 2023. [Електронний ресурс]. – Режим доступу: https://www.digitaltrends.com/ computing/stable-diffusion-pc-system-requirements/

38. OpenVivo. Text-to-Image Generation with Stable Diffusion v2 and OpenVINO – 2023. [Електронний ресурс]. – Режим доступу: https://docs.openvino.ai/2023.2/notebooks/236-stable-diffusion-v2-text-to-image-with-output.html#u-net

39. Novita.Ai. Stable Diffusion Img2Img Tutorial: Mastering the Technique - 2023. [Електронний ресурс]. – Режим доступу: https://medium.com/@novita_ai/stable-diffusion-img2img-tutorial-mastering-the-technique-52b391e5be40

40. Офіційна документація від HuggingFace. Text-to-image. [Електронний ресурс]. – Режим доступу: https://huggingface.co/runwayml/stable-diffusion-inpainting

41. Lvmin Zhang, Anyi Rao, Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models – 2023. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2302.05543

42. Офіційна документація Google Colab. Colab Enterprise. [Електронний ресурс]. – Режим доступу: https://cloud.google.com/colab/docs/introduction

43. Офіційна документація від Automatic1111. Stable Diffusion WebUI. [Електронний ресурс]. – Режим доступу: https://github.com/AUTOMATIC1111/stable-diffusion-webui

44. Chitwan Saharia, William Chan, Saurabh Saxena. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding – 2022. [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/2205.11487.pdf

45. CompVis. Офіційна документація до моделей Stable Diffusion v1.1 – v1.5 – 2022. [Електронний ресурс]. – Режим доступу: https://huggingface.co/CompVis

46. StabilityAI. Офіційна документація до моделей Stable Diffusion – 2022. [Електронний ресурс]. – Режим доступу: https://huggingface.co/stabilityai

47. LAION. Офіційна документація до тренувальних наборів даних – 2022. [Електронний ресурс]. – Режим доступу: https://laion.ai/

ДОДАТОК А

ЛІСТИНГ ПРОГРАМИ

Файл DiffusionLatentModelsInferenceText-to-Image.ipynb

```
import torch
torch_device = "cuda" if torch.cuda.is_available() else "cpu"

!pip install diffusers==0.11.1

!pip install transformers scipy ftfy accelerate

from transformers import CLIPTextModel, CLIPTokenizer
from diffusers import AutoencoderKL, UNet2DConditionModel,
PNDMScheduler

# 1. Load the autoencoder model which will be used to decode
the latents into image space.
vae = AutoencoderKL.from_pretrained("CompVis/stable-
diffusion-v1-4", subfolder="vae")

# 2. Load the tokenizer and text encoder to tokenize and encode
the text.
tokenizer = CLIPTokenizer.from_pretrained("openai/clip-vit-
large-patch14")
text_encoder = CLIPTextModel.from_pretrained("openai/clip-
vit-large-patch14")

# 3. The UNet model for generating the latents.
unet = UNet2DConditionModel.from_pretrained("CompVis/stable-
diffusion-v1-4", subfolder="unet")

from diffusers import LMSDiscreteScheduler

scheduler =
LMSDiscreteScheduler.from_pretrained("CompVis/stable-diffusion-v1-
4", subfolder="scheduler")

# move the models to the GPU.
vae = vae.to(torch_device)
text_encoder = text_encoder.to(torch_device)
unet = unet.to(torch_device)

#define the parameters we'll use to generate images.
prompt = ["a photograph of an astronaut riding a horse"]

height = 512                        # default height of Stable
Diffusion
width = 512                         # default width of Stable
Diffusion
```

```
      num_inference_steps = 100              # Number of denoising
steps
      guidance_scale = 7.5                 # Scale for SFG
      generator = torch.manual_seed(32)   # Seed generator to create
the inital latent noise

      batch_size = 1

      #get the text_embeddings for the prompt. These embeddings will
be used to condition the UNet model.
      text_input    =    tokenizer(prompt,    padding="max_length",
max_length=tokenizer.model_max_length,            truncation=True,
return_tensors="pt")

      with torch.no_grad():
        text_embeddings                                              =
text_encoder(text_input.input_ids.to(torch_device))[0]

      #get the unconditional text embeddings for classifier-free
guidance,
      #which are just the embeddings for the padding token (empty
text).
      #They need to have the same shape as the conditional
text_embeddings (batch_size and seq_length)
      max_length = text_input.input_ids.shape[-1]
      uncond_input = tokenizer(
          [""]       *       batch_size,       padding="max_length",
max_length=max_length, return_tensors="pt"
      )
      with torch.no_grad():
        uncond_embeddings                                            =
text_encoder(uncond_input.input_ids.to(torch_device))[0]

      #combining unconditional guidance and conditioned input
      text_embeddings        =        torch.cat([uncond_embeddings,
text_embeddings])

      #Generate the intial random noise.
      latents = torch.randn(
        (batch_size, unet.in_channels, height // 8, width // 8),
        generator=generator,
      )
      latents = latents.to(torch_device)

      #check the size of the latents
      latents.shape

      #initializing the scheduler
      scheduler.set_timesteps(num_inference_steps)

      #The K-LMS scheduler needs to multiply the latents by its sigma
values.
      latents = latents * scheduler.init_noise_sigma
```

```
      #the denoising loop
      from tqdm.auto import tqdm
      from torch import autocast

      for t in tqdm(scheduler.timesteps):
        # expand the latents if we are doing classifier-free guidance
to avoid doing two forward passes.
        latent_model_input = torch.cat([latents] * 2)

        latent_model_input                                          =
scheduler.scale_model_input(latent_model_input, t)

        # predict the noise residual
        with torch.no_grad():
          noise_pred      =      unet(latent_model_input,      t,
encoder_hidden_states=text_embeddings).sample

        # perform guidance
        noise_pred_uncond, noise_pred_text = noise_pred.chunk(2)
        noise_pred   =   noise_pred_uncond   +   guidance_scale   *
(noise_pred_text - noise_pred_uncond)

        # compute the previous noisy sample x_t -> x_t-1
        latents = scheduler.step(noise_pred, t, latents).prev_sample

      # scale and decode the image latents with vae
      latents = 1 / 0.18215 * latents

      with torch.no_grad():
        image = vae.decode(latents).sample

      # convert the image to PIL
      image = (image / 2 + 0.5).clamp(0, 1)
      image = image.detach().cpu().permute(0, 2, 3, 1).numpy()
      images = (image * 255).round().astype("uint8")
      pil_images = [Image.fromarray(image) for image in images]
      pil_images[0]
```

Файл LatentDiffusionModelTrainingText-to-Image,onLaion-400MDataset.ipynb

```
      #Installation
      !git clone https://github.com/crowsonkb/latent-diffusion.git
      !git clone https://github.com/CompVis/taming-transformers
      !pip install -e ./taming-transformers
      !pip install omegaconf>=2.0.0 pytorch-lightning>=1.0.8 torch-
fidelity einops
      !pip install transformers
      !pip install open_clip_torch
      !pip install autokeras
      !pip install tensorflow
      import sys
      sys.path.append(".")
      sys.path.append('./taming-transformers')
```

```
      from taming.models import vqgan

      #Download model
      %cd /content/latent-diffusion

      import os
      if
os.path.isfile(f"{model_path}/latent_diffusion_txt2img_f8_large.ck
pt"):
          print("Using saved model from Google Drive")
      else:
          !wget                                                    -O
$model_path/latent_diffusion_txt2img_f8_large.ckpt   https://ommer-
lab.com/files/latent-diffusion/nitro/txt2img-f8-large/model.ckpt

      #loading utils
      import torch
      from omegaconf import OmegaConf

      from ldm.util import instantiate_from_config

      #Import pyhton libraries
      import argparse, os, sys, glob
      import torch
      import numpy as np
      from omegaconf import OmegaConf
      from PIL import Image
      from tqdm.auto import tqdm, trange
      tqdm_auto_model = __import__("tqdm.auto", fromlist=[None])
      sys.modules['tqdm'] = tqdm_auto_model
      from einops import rearrange
      from torchvision.utils import make_grid
      import transformers
      import gc
      from ldm.util import instantiate_from_config
      from ldm.models.diffusion.ddim import DDIMSampler
      from ldm.models.diffusion.plms import PLMSSampler
      from open_clip import tokenizer
      import open_clip
      import tensorflow as tf

      #Load necessary functions


      def load_safety_model(clip_model):
          """load the safety model"""
          import autokeras as ak  # pylint: disable=import-outside-
toplevel
          from tensorflow.keras.models import load_model  # pylint:
disable=import-outside-toplevel
          from os.path import expanduser  # pylint: disable=import-
outside-toplevel
```

```
        home = expanduser("~")

        cache_folder = home + "/.cache/clip_retrieval/" +
clip_model.replace("/", "_")
        if clip_model == "ViT-L/14":
            model_dir = cache_folder +
"/clip_autokeras_binary_nsfw"
            dim = 768
        elif clip_model == "ViT-B/32":
            model_dir = cache_folder + "/clip_autokeras_nsfw_b32"
            dim = 512
        else:
            raise ValueError("Unknown clip model")
        if not os.path.exists(model_dir):
            os.makedirs(cache_folder, exist_ok=True)

            from urllib.request import urlretrieve  # pylint:
disable=import-outside-toplevel

            path_to_zip_file = cache_folder +
"/clip_autokeras_binary_nsfw.zip"
            if clip_model == "ViT-L/14":
                url_model =
"https://raw.githubusercontent.com/LAION-AI/CLIP-based-NSFW-
Detector/main/clip_autokeras_binary_nsfw.zip"
            elif clip_model == "ViT-B/32":
                url_model = (
                    "https://raw.githubusercontent.com/LAION-
AI/CLIP-based-NSFW-Detector/main/clip_autokeras_nsfw_b32.zip"
                )
            else:
                raise ValueError("Unknown model
{}".format(clip_model))
            urlretrieve(url_model, path_to_zip_file)
            import zipfile  # pylint: disable=import-outside-
toplevel

            with zipfile.ZipFile(path_to_zip_file, "r") as
zip_ref:
                zip_ref.extractall(cache_folder)

        loaded_model = load_model(model_dir,
custom_objects=ak.CUSTOM_OBJECTS)
        loaded_model.predict(np.random.rand(10 ** 3,
dim).astype("float32"), batch_size=10 ** 3)

        return loaded_model

    def is_unsafe(safety_model, embeddings, threshold=0.5):
        """find unsafe embeddings"""
        nsfw_values = safety_model.predict(embeddings,
batch_size=embeddings.shape[0])
        x = np.array([e[0] for e in nsfw_values])
```

```
            #print(x)
            return True if x > threshold else False
        #NSFW CLIP Filter
        safety_model = load_safety_model("ViT-B/32")
        clip_model,             _,            preprocess            =
open_clip.create_model_and_transforms('ViT-B-32',
pretrained='openai')

        def load_model_from_config(config, ckpt, verbose=False):
            print(f"Loading model from {ckpt}")
            pl_sd = torch.load(ckpt, map_location="cuda:0")
            sd = pl_sd["state_dict"]
            model = instantiate_from_config(config.model)
            m, u = model.load_state_dict(sd, strict=False)
            if len(m) > 0 and verbose:
                print("missing keys:")
                print(m)
            if len(u) > 0 and verbose:
                print("unexpected keys:")
                print(u)

            model = model.half().cuda()
            model.eval()
            return model

        config   =   OmegaConf.load("configs/latent-diffusion/txt2img-
1p4B-eval.yaml")
        model             =             load_model_from_config(config,
f"{model_path}/latent_diffusion_txt2img_f8_large.ckpt")

        device  =  torch.device("cuda")  if  torch.cuda.is_available()
else torch.device("cpu")
        model = model.to(device)
        def run(opt):
            torch.cuda.empty_cache()
            gc.collect()
            if opt.plms:
                opt.ddim_eta = 0
                sampler = PLMSSampler(model)
            else:
                sampler = DDIMSampler(model)

            os.makedirs(opt.outdir, exist_ok=True)
            outpath = opt.outdir

            prompt = opt.prompt


            sample_path = os.path.join(outpath, "samples")
            os.makedirs(sample_path, exist_ok=True)
            base_count = len(os.listdir(sample_path))

            all_samples=list()
```

```
          with torch.no_grad():
              with torch.cuda.amp.autocast():
                  with model.ema_scope():
                      uc = None
                      if opt.scale > 0:
                          uc                                          =
model.get_learned_conditioning(opt.n_samples * [""])
                      for n in trange(opt.n_iter, desc="Sampling"):
                          c                                           =
model.get_learned_conditioning(opt.n_samples * [prompt])
                          shape = [4, opt.H//8, opt.W//8]
                          samples_ddim,             _                 =
sampler.sample(S=opt.ddim_steps,

conditioning=c,

batch_size=opt.n_samples,

shape=shape,

verbose=False,

unconditional_guidance_scale=opt.scale,

unconditional_conditioning=uc,

eta=opt.ddim_eta)

                          x_samples_ddim                              =
model.decode_first_stage(samples_ddim)
                          x_samples_ddim                              =
torch.clamp((x_samples_ddim+1.0)/2.0, min=0.0, max=1.0)

                          for x_sample in x_samples_ddim:
                              x_sample        =       255.        *
rearrange(x_sample.cpu().numpy(), 'c h w -> h w c')
                              image_vector                            =
Image.fromarray(x_sample.astype(np.uint8))
                              image                                   =
preprocess(image_vector).unsqueeze(0)
                              with torch.no_grad():
                                image_features                        =
clip_model.encode_image(image)
                              image_features                         /=
image_features.norm(dim=-1, keepdim=True)
                              query                                   =
image_features.cpu().detach().numpy().astype("float32")
                              unsafe                                  =
is_unsafe(safety_model,query,opt.nsfw_threshold)
                              if(not unsafe):

image_vector.save(os.path.join(sample_path,
f"{base_count:04}.png"))
```

```
                                else:
                                  raise Exception('Potential NSFW
content was detected on your outputs. Try again with different
prompts. If you feel your prompt was not supposed to give NSFW
outputs, this may be due to a bias in the model')
                                base_count += 1
                            all_samples.append(x_samples_ddim)


          # additionally, save as grid
          grid = torch.stack(all_samples, 0)
          grid = rearrange(grid, 'n b c h w -> (n b) c h w')
          grid = make_grid(grid, nrow=opt.n_samples)

          # to image
          grid = 255. * rearrange(grid, 'c h w -> h w
c').cpu().numpy()


Image.fromarray(grid.astype(np.uint8)).save(os.path.join(outpath,
f'{prompt.replace(" ", "-")}.png'))
          display(Image.fromarray(grid.astype(np.uint8)))
          #print(f"Your samples are ready and waiting four you here:
\n{outpath} \nEnjoy.")


      #Generation parameters
      import argparse
      Prompt = "A mecha robot holding a sign that reads: 'Is AI art,
art?" #@param{type:"string"}
      Steps = 50 #@param {type:"integer"}
      ETA = 0.0 #@param{type:"number"}
      Iterations = 2 #@param{type:"integer"}
      Width=256 #@param{type:"integer"}
      Height=256 #@param{type:"integer"}
      Samples_in_parallel=3 #@param{type:"integer"}
      Diversity_scale=5.0 #@param {type:"number"}
      PLMS_sampling=True #@param {type:"boolean"}

      args = argparse.Namespace(
          prompt = Prompt,
          outdir=f'{outputs_path}',
          ddim_steps = Steps,
          ddim_eta = ETA,
          n_iter = Iterations,
          W=Width,
          H=Height,
          n_samples=Samples_in_parallel,
          scale=Diversity_scale,
          plms=PLMS_sampling,
          nsfw_threshold=0.5
      )
      run(args)
```

ДОДАТОК Б

ВІДГУК КЕРІВНИКА

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ДНІПРОВСЬКА ПОЛІТЕХНІКА»

Факультет інформаційних технологій
Кафедра програмного забезпечення комп'ютерних систем

ВІДГУК

Наукового керівника Спірінцева В'ячеслава Василійовича, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання, посада, місце роботи)

на магістерську роботу

студента Ніколаєнка Артема Віталійовича
(прізвище, ім'я, по батькові)

курсу II групи 122м-22-3
спеціальності 122 Комп'ютерні науки
освітньої програми
на тему Дослідження та застосування методу Stable Diffusion на базі штучного інтелекту в контексті генеративного мистецтва

Актуальність теми Представлена магістерська кваліфікаційна робота присвячена дослідженню та використанню моделей латентної дифузії у завданнях генеративного мистецтва. На сьогоднішній день машинне навчання та генеративне мистецтво є провідними та стрімко зростаючими галузями. Використання моделей латентної дифузії наразі є одним із найпоширеніших підходів до вирішення завдань генеративного мистецтва, який характеризується своєю якістю та ефективністю, тому з огляду на це, робота характеризується своєю актуальністю та своєчасністю.

Мета досліджень Полягає у дослідженні та вдосконаленні якості генерованих зображень і ефективності налаштування параметрів генерації при використанні моделей латентної дифузії, як інструменту у завданнях генеративного мистецтва.

Коротка характеристика розділів роботи Перший розділ роботи містить аналітичний огляд існуючих методів та підходів до вирішення завдань генеративного мистецтва за темою магістерської роботи. Другий розділ присвячено детальному дослідженню та аналізу моделей латентної дифузії як засобу для вирішення завдань генеративного мистецтва. У третьому розділі розглянуто практичне використання моделей латентної дифузії, їх параметрів генерації та вплив значень даних на якість та ефективність створення зображень, на основі отриманих результатів було сформовано перелік рекомендацій та вказівок для оптимального налаштування параметрів генерації зображень при роботі із моделями латентної дифузії у завданнях генеративного мистецтва.

Практичне значення роботи Отримані результати роботи є застосовними при використанні моделей латентної дифузії для генерації зображень у завданнях генеративного мистецтва. Дані результати дозволяють підвищити якість генерованих зображень та оптимізувати процес підбору параметрів генерації, збільшивши ефективність процесу генерації та ефективність використання задіяних обчислювальних ресурсів.

Зауваження та недоліки В роботі відсутній огляд використання моделей латентної дифузії у інших сферах генеративного мистецтва, окрім генерації зображень, наприклад створення анімації, музики або відео. Також у роботі не було запропоновано рекомендацій щодо покращення технічних або програмних аспектів роботи моделі, основну увагу було приділено більш ефективному використанню даних моделей у їх поточному вигляді.

Висновки та оцінка Магістром було проведено аналіз та порівняння можливих методологій та підходів до розв’язання задач генеративного мистецтва та обрано оптимальний варіант, який дозволяє створення зображень високої якості та ефективність використання обчислювальних ресурсів. Під час виконання магістерської кваліфікаційної роботи студент Ніколаєнко А.В. проявив себе грамотним, кваліфікованим спеціалістом, здатним самостійно

приймати складні технічні рішення. Вважаю, що магістерська кваліфікаційна робота заслуговує оцінку 95 «відмінно», а Ніколаєнко А.В. – присвоєння кваліфікації «магістра» з комп'ютерних наук.

Науковий керівник Спірінцев В.В., доцент
(прізвище, ім'я, по батькові, посада, місце роботи)

«_____»_____20___р. _____
(підпис)

Додаток В

РЕЦЕНЗІЯ
на кваліфікаційну роботу

студента Ніколаєнка Артема Віталійовича
(прізвище, ім'я, по батькові)

курсу II групи 122м-22-3
кафедри програмного забезпечення комп'ютерних систем
спеціальності 122 Комп'ютерні науки
Тема роботи Дослідження та застосування методу Stable Diffusion на базі штучного інтелекту в контексті генеративного мистецтва

Стисла характеристика розділів роботи Перший розділ містить огляд існуючих методів та підходів до вирішення завдань генеративного мистецтва, переваги та недоліки різних методів, вибір методу для магістерської роботи та обґрунтування його доцільності. Другий розділ присвячено детальному дослідженню моделей латентної дифузії, алгоритмам роботи та навчання даних моделей, перевагам та недолікам даних моделей та параметрів генерації, що використовуються при роботі даного класу генеративних моделей. У третьому розділі було наведено практичне використання моделей латентної дифузії та аналіз отриманих результатів для виявлення впливу значень параметрів генерації на якість та ефективність генерації зображень.

Пропозиції, внесені студентом, рівень їх наукового обґрунтування В даній кваліфікаційній роботі студентом надано пропозицій щодо вирішення поставлених задач. Кожна з пропозицій була обґрунтована та підкріплена теоретичними та практичними даними.

Практичне значення роботи Результати роботи можуть бути застосовані для подальших наукових досліджень в даній сфері, а також вони можуть бути корисними для практичного використання у завданнях генеративного мистецтва.

Якість оформлення роботи Магістерська кваліфікаційна робота, яку подано на рецензію, виконана у повному обсязі у встановлений термін. Робота є добре структурованою та достатньо проілюстрованою. Викладена основна суть проблеми, що вирішується в ході виконання роботи, і шляхів її вирішення.

Недоліки в роботі Відсутність більш детального аналізу використання моделей латентної дифузії у інших завданнях генеративного мистецтва, а не тільки у сфері генерації зображень. Проте вказаний недолік не впливає на позитивне враження від роботи.

Загальний висновок Магістерська кваліфікаційна робота виконана у відповідності з завданням із дотриманням всіх вимог.

Оцінка магістерської роботи Робота заслуговує оцінки 95 «відмінно», а студент Ніколаєнко А.В. – присвоєння кваліфікації «магістра» з комп'ютерних наук.

Рецензент _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання, посада, місце роботи)

«_____»_____20___р. _____
(підпис)

ДОДАТОК Г

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім’я файла	**Опис**
Пояснювальні документи	
Диплом_Ніколаєнко А.В.doc	Пояснювальна записка роботи. Документ Word.
Диплом_Ніколаєнко А.В.pdf	Пояснювальна записка роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди скриптів Notebook Jupiter.
Презентація	
Презентація_Ніколаєнко.pptx	Презентація роботи