НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.8

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«___»_____2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою
«Інформаційні управляючі системи та технології»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Інтелектуальна система генерації зображень з текстових представлень на базі ймовірнісних моделей дифузії»

Виконав:
студент 2 курсу, групи ІС-32мп
Мулік Рустам Юрійович _____

Керівник:
доцент каф. ІСТ, к.т.н., доц.
Цьопа Наталія Володимирівна _____

Рецензент:
зав. кафедри ІСТ ФІТ
КНУ ім. Тараса Шевченка, д.т.н., проф.
Дружинін Володимир Анатолійович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____2024 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Муліку Рустаму Юрійовичу

1. Тема дисертації «Інтелектуальна система генерації зображень з текстових представлень на базі ймовірнісних моделей дифузій», науковий керівник дисертації Цьопа Наталія Володимирівна, к.т.н., доц., затверджені наказом по університету від «08» 11 2024 р. № 5016-с

2. Термін подання студентом дисертації «09» 12 2024 р.

3. Об'єкт дослідження: процес генерації зображень на основі текстових представлень з використанням дифузійних моделей.

4. Вихідні дані: час відповіді до 15 секунд для генерації попереднього перегляду зображення (при використанні CPU, без GPU); підтримка форматів 512x512 та 1024x1024 пікселів; захист від зловмисного використання (фільтрація заборонених запитів); можливість обробки коротких і довгих описів з дотриманням семантичної відповідності.

5. Перелік завдань, які потрібно розробити: провести аналіз предметної області дослідження, систематизувати існуючі архітектури моделей перетворення текстових даних в зображення, спроектувати систему генерації зображень з текстових даних з використанням моделі ймовірнісної дифузії, інтегрувати розроблену модель в графічний інтерфейс для взаємодії з користувачем.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: діаграма варіантів використання (А3), діаграма компонентів (А3), алгоритм роботи системи (А3), діаграма діяльності (А3), діаграма послідовності (А3), архітектура нейронної мережі (А3), схема станів (А3), архітектура генеративно-змагальної мережі (А3).

7. Орієнтовний перелік публікацій: публікації відсутні.

8. Дата видачі завдання 02.09.2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Огляд та аналіз існуючих рішень	10.09.24	виконано
2.	Визначення цілей та задач розробки	17.09.24	виконано
3.	Опис предметного середовища	17.09.24	виконано
4.	Визначення вхідних та вихідних даних;	24.09.24	виконано
5.	Визначення методів і засобів для вирішення завдання	24.09.24	виконано
6.	Розробка технічного завдання на систему	30.09.24	виконано
7.	Детальне проектування елементів системи	30.09.24	виконано
8.	Опис структури ймовірнісної моделі дифузії	07.10.24	виконано
9.	Підготовка набору даних тексту та зображень	07.10.24	виконано
10.	Розробка архітектури системи	18.10.24	виконано

Студент Рустам МУЛІК

Науковий керівник Наталія ЦЬОПА

РЕФЕРАТ

Інтелектуальна система генерації зображень з текстових представлень на базі ймовірнісних моделей дифузій: 105 с., 23 табл., 21 рис., 8 дод., 31 джерело.

Об'єктом дослідження є процес генерації зображень на основі текстових представлень з використанням дифузійних моделей.

Предметом дослідження є алгоритмічні принципи та архітектурні особливості ймовірнісних дифузійних моделей, які дозволяють досягати високої точності та якості генерованих зображень на основі текстових описів.

Метою магістерської дисертації є підвищення ефективності створення візуальних матеріалів шляхом автоматизації цього процесу за допомогою розробки інтелектуальної системи генерації зображень з текстових представлень на основі ймовірнісних моделей дифузії.

У сучасному світі цифрові технології відіграють ключову роль у всіх сферах діяльності, а автоматизація творчих процесів стає все більш затребуваною. Генерація зображень на основі текстових описів є перспективною технологією, що знаходить застосування в медіа, маркетингу, освіті, дизайні та інших галузях. Особливу увагу привертають дифузійні моделі, які завдяки ймовірнісним підходам забезпечують високу якість, деталізацію та реалістичність зображень, дозволяючи візуалізувати складні концепції з тексту. Це сприяє автоматизації рутинних процесів, підвищенню креативності та зниженню витрат у комерційних та освітніх проєктах, а також відповідає сучасним тенденціям зростання ролі ШІ в глобальній економіці.

Ключові слова: ТЕКСТОВІ ПРЕДСТАВЛЕННЯ, ДИФУЗІЙНІ МОДЕЛІ, ЙМОВІРНІСНІ МОДЕЛІ, ШТУЧНИЙ ІНТЕЛЕКТ, ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ, ГЛИБИННЕ НАВЧАННЯ, АЛГОРИТМИ МАШИННОГО НАВЧАННЯ, МУЛЬТИМОДАЛЬНИЙ СИНТЕЗ, PDM.

ABSTRACT

Intelligent system for generating images from text representations based on probabilistic diffusion models: 105 p., 23 tables, 21 figures, 8 appendices, 31 sources.

The object of the study is the process of generating images based on text representations using diffusion models.

The subject of the study is the algorithmic principles and architectural features of probabilistic diffusion models, which allow achieving high accuracy and quality of generated images based on text descriptions.

The purpose of the master's thesis is to develop an intelligent system for generating images from text representations based on probabilistic diffusion models to automate the process of creating visual materials.

In the modern world, digital technologies play a key role in all areas of activity, and the automation of creative processes is becoming increasingly popular. Generating images based on text descriptions is a promising technology that finds application in media, marketing, education, design and other industries. Of particular interest are diffusion models, which, thanks to probabilistic approaches, provide high quality, detail and realism of images, allowing to visualize complex concepts from text. This contributes to the automation of routine processes, increased creativity and reduced costs in commercial and educational projects, and also corresponds to the current trends of increasing the role of AI in the global economy.

Keywords: TEXT REPRESENTATIONS, DIFFUSION MODELS, PROBABILITY MODELS, ARTIFICIAL INTELLIGENCE, INTELLIGENT SYSTEMS, DEEP LEARNING, MACHINE LEARNING ALGORITHMS, MULTIMODAL SYNTHESIS, PDM.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

A2D – Attention-to-Diffusion – механізм уваги до дифузійного процесу, що забезпечує адаптивну передачу ознак

DDSM – Dynamic Diffusion Sampling Mechanism – динамічний механізм вибірки для дифузійних моделей, що оптимізує процес вибору даних

DDPM – Denoising Diffusion Probabilistic Model – ймовірнісна модель дифузії з усуненням шуму, що моделює послідовний процес усунення шуму

DDIM – Denoising Diffusion Implicit Model – неявна модель дифузії з усуненням шуму, що скорочує кількість етапів генерації без втрати якості

DCGAN – Deep Convolutional Generative Adversarial Network − глибока згорткова генеративна змагальна мережа

Relu – Rectified Exponential Linear Unit – зрізаний лінійний вузол

CNN – Convolutional Neural Network − згорткова нейронна мережа

VAE – Variational Autoencoder − варіаційний автокодувальник

CV – Computer Vision − комп'ютерний зір

DiT – Diffusion Transformer − дифузійний трансформер

PNDM – Pseudo Numerical Diffusion Model − псевдонумерична дифузійна модель

ВСТУП

Актуальність дослідження полягає в тому, що у сучасному світі, де цифрові технології стали невід'ємною частиною усіх сфер життя, відбувається інтенсивний розвиток інструментів, які дозволяють автоматизувати творчі та аналітичні процеси. Генерація зображень на основі текстових описів є однією з найперспективніших технологій, здатних трансформувати підхід до створення контенту у галузях медіа, маркетингу, освіти, дизайну, архітектури, а також у науці. Це обумовлено глобальними потребами в інструментах, які поєднують високу продуктивність, точність та можливість швидкої адаптації до індивідуальних потреб користувачів.

Особливу роль у цьому контексті відіграють дифузійні моделі, які ґрунтуються на складних ймовірнісних процесах, що дозволяють поступово покращувати якість генерованих зображень. Завдяки здатності цих моделей ефективно працювати з шумовими даними, вони забезпечують плавний перехід від випадкової початкової конфігурації до детально опрацьованого зображення, зберігаючи при цьому інформацію, закладену у текстовому описі.

Це дозволяє отримувати візуальний контент із високою роздільною здатністю, багатими деталями та реалістичними текстурами, що неможливо досягти за допомогою традиційних методів генерації. Крім того, дифузійні моделі демонструють відмінні результати у завданнях, пов’язаних із комплексною інтерпретацією багатозначних або художньо складних текстових описів, що відкриває нові горизонти у сфері інтерактивного дизайну та творчості. Значущість розробки таких систем обумовлена також попитом на автоматизацію рутинних процесів у комерційній сфері, зокрема у виробництві рекламних матеріалів, створенні презентацій, розробці навчальних курсів тощо. Використання генеративних моделей у цих процесах не лише знижує витрати, але й підвищує креативність завдяки швидкому створенню різноманітних варіантів візуалізації. У наукових дослідженнях та освітніх проєктах системи такого типу дозволяють наочно демонструвати складні концепції, що суттєво полегшує сприйняття

інформації та сприяє кращому засвоєнню знань. Крім того, актуальність використання дифузійних моделей посилюється на фоні тенденцій до зростання ролі штучного інтелекту у глобальній економіці. Їх застосування у таких сферах, як створення цифрового мистецтва, розробка відеоігор або створення контенту для соціальних мереж.

Метою магістерської дисертації є підвищення ефективності створення візуальних матеріалів шляхом автоматизації цього процесу за допомогою розробки інтелектуальної системи генерації зображень з текстових представлень на основі ймовірнісних моделей дифузії.

Для досягення мети магістерської дисертації необхідно виконати наступні **завдання**:

– Дослідити теоретичні аспекти використання генеративних моделей для генерації зображень, а також існуючі рішення.

– Систематизувати існуючі архітектури моделей перетворення текстових даних в зображення.

– Спроектувати систему генерації зображень з використанням моделі ймовірнісної дифузії.

– Розробити програмне забезпечення для системи генерації зображень з використанням моделі ймовірнісної дифузії.

– Провести тестування функціональних можливостей генерації зображень з використанням моделі ймовірнісної дифузії.

– Розробити стартап-проєкт з визначення потенційних стратегій виходу розробленої системи на ринок.

Об'єктом дослідження є процес генерації зображень на основі текстових представлень з використанням дифузійних моделей.

Предметом дослідження є алгоритмічні принципи та архітектурні особливості ймовірнісних дифузійних моделей, які дозволяють досягати високої точності та якості генерованих зображень на основі текстових описів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

Генеративні моделі представляють собою потужний інструмент для вивчення різних типів розподілів даних, який базується на принципах неконтрольованого навчання та дозволяє модельним архітектурам відтворювати основні закономірності складних розподілів. Головною метою генеративних моделей є глибоке опанування латентного розподілу вхідних даних, що дозволяє формувати нові точки даних із заданими варіаціями, що водночас зберігають суттєві статистичні властивості оригінального розподілу. Такий підхід забезпечує можливість створення синтетичних даних, які є репрезентативними та здатні імітувати структуру реальних даних у різних доменах, зокрема візуальних і текстових. Сучасні генеративні моделі, такі як варіаційні автоенкодери (VAE), генеративні змагальні мережі (GAN) та дифузійні моделі, кожна з яких використовує різні підходи до моделювання латентного простору, орієнтовані на побудову функціональних апроксимацій для складних розподілів даних. Ці архітектури забезпечують можливість синтезу нового контенту шляхом навчання на багатомодальних розподілах і включають механізми, що мінімізують ентропію відтворення при збереженні ключових семантичних характеристик. Особливої уваги заслуговує використання генеративних моделей для задачі перетворення текстових описів у візуальні образи. Цей процес є надзвичайно складним, оскільки передбачає семантичну декодування природної мови та конвертацію її контекстуальних особливостей у візуальні векторні простори.

1.1 Огляд існуючих рішень

Генеративні моделі для перетворення тексту на зображення — це моделі машинного навчання, здатні створювати зображення на основі описів, поданих природною мовою. Такі моделі можуть генерувати візуальний контент, що точно відповідає заданому опису, наприклад, зображення кота в капелюсі. За останні роки точність і реалістичність цих моделей значно зросли завдяки досягненням у галузі

глибоких нейронних мереж, дифузійних моделей, доступності великих наборів даних та високопродуктивних обчислювальних ресурсів. У результаті згенеровані зображення стають дедалі реалістичнішими. У наступних розділах розглядаються основні сучасні моделі, які використовуються для задач перетворення тексту на зображення.

1.1.1 Imagen

Imagen – це генеративна модель для створення зображень на основі текстових описів, розроблена Google Research. Її архітектура ґрунтується на інтеграції великих мовних моделей із дифузійними процесами, що дозволяє досягти високого рівня якості та семантичної відповідності між текстом і зображенням. Основною технічною інновацією Imagen є глибока мультимодальна взаємодія, яка базується на попередньо навчених мовних моделях для точного вилучення семантичних репрезентацій тексту та подальшого використання їх у генеративному процесі. Архітектура Imagen складається з кількох ключових компонентів, які будуть описані нижче.

Перший етап – це текстовий енкодер, який використовує попередньо навчену мовну модель, таку як T5 (Text-to-Text Transfer Transformer), для отримання латентного представлення тексту. Завдяки цьому система враховує контекст, складні залежності та семантичні нюанси текстового запиту.

Далі латентні репрезентації тексту слугують основою для умовного дифузійного процесу. Генеративна частина Imagen побудована на основі каскадної дифузійної моделі. Основна ідея каскадного підходу полягає у поступовому підвищенні роздільної здатності зображень через кілька етапів, де кожен етап використовує окрему дифузійну модель.

Перший етап генерує низькорозмірне зображення (наприклад, 64x64 пікселів), яке потім проходить через каскад суперрезолюційних моделей, поступово досягаючи високої роздільної здатності (наприклад, 1024x1024 пікселів). Це забезпечує ефективне використання обчислювальних ресурсів і

дозволяє моделі фокусуватися на деталях зображення лише на останніх етапах генерації. Ключовою особливістю Imagen є інтеграція текстово-візуальної симбіозної архітектури, що використовує CLIP-подібні механізми для вирівнювання текстових і візуальних репрезентацій.

На рисунку 1.1 представлений приклад зображення з використанням моделі Imagen за наступним текстовим описом: “Старий дерев'яний маяк на узбережжі під час заходу сонця, з хвилями, що б'ються об скелі. Атмосфера – спокійна, з відтінками помаранчевого та рожевого”.

Рисунок 1.1 – Зображення, згенероване моделлю Imagen

1.1.2 Stable Diffusion

Stable Diffusion – це сучасна генеративна модель, призначена для високоякісного синтезу зображень на основі текстових описів, що реалізує процес дифузії для поступового вдосконалення зображення. Побудована на основі ймовірнісних дифузійних моделей, ця технологія використовує процеси шумування та зворотного відновлення, де модель додає випадковий шум до зображення, а потім навчається поступово відновлювати зображення з цього шуму, ґрунтуючись на текстовому описі. Stable Diffusion була розроблена з акцентом на ефективність обчислень, що дозволяє знизити вимоги до апаратного забезпечення та забезпечити високий рівень деталізації з меншою затратою ресурсів. Архітектура Stable Diffusion базується на використанні латентного простору, який значно зменшує розмірність зображення, що дозволяє моделі працювати з високою швидкістю.

Спершу текстовий опис кодується в текстовому енкодері на основі трансформера, що формує багатовимірне представлення змісту опису. Це представлення потім інтегрується у процес генерації зображення, де модель генерує зображення в латентному просторі, а не в реальних координатах пікселів, що значно оптимізує обчислювальний процес. Stable Diffusion проходить через кілька послідовних етапів дифузії, де кожен крок наближає зображення до реалістичного, зменшуючи шум і збагачуючи його деталями, що відповідають текстовому опису. На кожному кроці моделі необхідно зробити припущення щодо правильного стану зображення, враховуючи шум і контекст опису, що дозволяє поступово уточнювати деталі.

Функція втрат у Stable Diffusion оптимізована для мінімізації відмінностей між поточним станом зображення і передбаченим станом на кожному етапі, використовуючи спеціальні методи для прогнозування та видалення шуму. Завдяки цій ітеративній оптимізації модель досягає високої реалістичності зображень, при цьому враховуючи навіть складні елементи, такі як освітлення, текстури та композиційні взаємозв’язки між об’єктами.

Особливістю Stable Diffusion є можливість налаштування глибини ітерацій, що дозволяє користувачам контролювати рівень деталізації та стилізації, обираючи між більш узагальненим і стилізованим зображенням або реалістичним, але ресурсомістким результатом.

На рисунку 1.2 представлений приклад зображення з використанням моделі SD XL за наступним описом: “Гіперреалістична сцену великого класичного коридору всередині розкішного палацу. Передпокій обшитий високими колонами та прикрашений вишуканими позолоченими картинами на стінах. Величезні океанські хвилі проносяться по коридору.”

Рисунок 1.2 – Зображення, згенероване Stable Diffusion

1.1.3 Runway ML

Runway ML – це сучасна генеративна платформа, орієнтована на створення візуального та мультимедійного контенту, що базується на інтеграції провідних моделей штучного інтелекту, включаючи дифузійні моделі та трансформерні архітектури. Runway ML забезпечує професіоналів і креаторів доступом до потужних інструментів для генерації зображень, редагування відео, створення анімацій та інших форм контенту, зокрема на основі текстових запитів. Архітектура Runway ML інтегрує декілька різнорівневих моделей, кожна з яких адаптована для специфічних завдань, таких як генерація реалістичних текстур, реконструкція зображень або редагування сцен з урахуванням семантичного контексту. Завдяки цьому платформа здатна обробляти складні текстові запити, декодувати багатовимірні текстові вектори, генерувати зображення з високим ступенем деталізації та візуальної складності, а також точно інтерпретувати атрибути, текстури, атмосферу, освітлення й композиційні аспекти.

Runway ML побудована на основі модульного принципу, що дозволяє інтегрувати її функціонал із широким спектром професійних програм і робочих процесів. Платформа використовує серверну обчислювальну інфраструктуру для виконання ресурсоємних обчислень, що дозволяє користувачам працювати навіть із менш потужними локальними пристроями. Завдяки використанню трансформерної архітектури і багатошарової уваги Runway ML оптимізує розподіл обчислювальних ресурсів на ключові елементи опису, забезпечуючи точне збереження контексту, стилістичних характеристик і емоційного настрою у створеному контенті. Платформа підтримує інтерактивний веб-інтерфейс, що дозволяє користувачам завантажувати вхідні дані (зображення, відео або текстові запити), налаштовувати параметри генерації та отримувати готовий результат у реальному часі. Крім того, Runway ML інтегрує функції співпраці, що робить її ідеальним вибором для творчих команд. Наприклад, користувачі можуть експортувати проєкти у форматах, сумісних із професійними інструментами

редагування, такими як Adobe After Effects чи Premiere Pro, або використовувати API для автоматизації робочих процесів.

На рисунку 1.3 представлений приклад зображення з використанням моделі Runway ML за наступним описом: “Гірське озеро на світанку, оточене сосновим лісом і вкрите легким туманом. Небо поступово змінюється від рожевого до блакитного, створюючи спокійну атмосферу”.

Рисунок 1.3 – Зображення, згенероване Runway ML

1.2 Порівняння систем перетворення тексту в зображення

В таблиці 1.1 представлено порівняльний аналіз вищерозглянутих систем перетворення тексту в зображення.

Таблиця 1.1 – Порівняльний аналіз систем перетворення текстових даних в зображення

Характеристика	**Imagen**	**Stable Diffusion**	**Runway ML**
Технологічна основа	Використовує мовну модель для аналізу тексту та каскадну дифузійну модель для створення зображень із семантичною відповідністю	Використовує дифузійну модель, яка поступово відновлює зображення з шуму, орієнтуючись на текстовий опис	Використовує сучасні моделі штучного інтелекту (включаючи дифузійні моделі) для створення зображень, редагування відео, а також інших креативних задач
Якість згенерованих зображень	Висока реалістичність і деталізація, особливо у складних текстурах і точній відповідності текстовому опису	Забезпечує високу якість із можливістю налаштування стилю та рівня деталізації	Генерує зображення з високою реалістичністю та підтримує інтеграцію з різними моделями, що забезпечує гнучкість у виборі стилю та рівня деталізації
Можливості налаштування	Забезпечує точний контроль над деталізацією та композицією через текстові інструкції завдяки вирівнюванню текстово-візуальних репрезентацій	Гнучкі можливості налаштування, що дозволяють змінювати стиль і деталізацію відповідно до текстових запитів	Пропонує широкий спектр налаштувань для створення зображень, інтеграція з інструментами робить процес адаптивним

Продовження таблиці 1.1

Характеристика	Imagen	Stable Diffusion	Runway ML
Обмеження на використання	Доступ обмежений, переважно через внутрішні дослідницькі ініціативи Google, комерційне використання недоступне	Вільний доступ для комерційного та особистого використання	Платформа доступна через підписку; дозволяє комерційне використання, орієнтована на професіоналів та креаторів контенту
Потреби в обчислювальних потужностях	Високі, потрібні потужні сервери для генерації зображень або доступ до інфраструктури Google Research	Підтримує роботу як на потужних серверах, так і локально на менш потужних пристроях	Обчислення виконуються на стороні сервера, тому вимоги до локальних ресурсів низькі; потрібен швидкий інтернет
Можливості для відтворення детальних сцен	Забезпечує високу деталізацію та точність у складних композиціях завдяки каскадній структурі дифузійної моделі	Підтримка складних сцен та багатокомпонентних зображень	Підтримує складні сцени з хорошим рівнем деталізації; інструменти дозволяють користувачам вручну покращувати створені зображення

Джерело: розроблено автором на основі [8].

На основі проведеного порівняльного аналізу в таблиці 1.1, можна зробити наступні висновки:

– Imagen вирізняється здатністю генерувати високоякісні, деталізовані зображення завдяки каскадній дифузійній архітектурі, яка дозволяє поступово підвищувати роздільну здатність зображення. Використання мовної моделі T5 забезпечує високу семантичну відповідність тексту та зображення, що робить її однією з провідних моделей для текстово-візуальної генерації. Хоча модель має

обмежений доступ і переважно використовується для дослідницьких цілей, її інноваційний підхід до текстово-візуального вирівнювання та висока реалістичність демонструють великий потенціал для майбутніх розробок.

– Stable Diffusion відзначається ефективністю і здатністю підтримувати багатокомпонентні, складні сцени з можливістю адаптації до різноманітних стилів і рівнів деталізації. Дифузійна архітектура дозволяє реалізувати гнучкий процес відновлення зображення з шуму, що надає користувачеві контроль над композиційними та стилістичними аспектами зображення. Завдяки відкритому доступу та можливості працювати на різних обчислювальних платформах, модель є гарним варіантом для комерційних і особистих проєктів, забезпечуючи масштабованість і адаптивність.

– Runway ML займає особливу позицію серед генеративних платформ завдяки своїй орієнтації на професійне використання і гнучкість у роботі з мультимедійним контентом. Використовуючи сучасні дифузійні моделі та трансформерну архітектуру, Runway ML забезпечує високий рівень точності у відтворенні складних сцен і контекстів. Основний акцент платформи спрямований на створення реалістичного та деталізованого контенту, що робить її придатною для широкого спектра завдань, включаючи генерацію відео, зображень і редагування контенту. Runway ML пропонує передбачувані результати та розширені можливості налаштування, що дозволяє досягати більшої відповідності текстовим описам і вимогам проєкту.

Проте кожна з цих моделей має свої недоліки, які варто розглянути. Imagen, хоч і забезпечує виняткову якість та деталізацію, має обмежений доступ, оскільки інтегрована в закриту екосистему Google Gemini, що ускладнює її використання для незалежних розробників та комерційних проєктів. Stable Diffusion, попри свою гнучкість і відкритий доступ, може вимагати значних обчислювальних ресурсів для створення високоякісних зображень, а також потребує налаштування, що може бути складним для новачків. Платформа Runway ML є платною, і доступ до повного функціоналу можливий лише через підписку, що може бути фінансово обтяжливим для окремих користувачів або невеликих команд. Оскільки обчислення

виконуються на серверній інфраструктурі, ефективність роботи платформи залежить від стабільного і швидкого інтернет-з’єднання, а у випадку технічних проблем чи високого навантаження на сервери можуть виникати затримки в обробці запитів. Runway ML не підтримує локальний режим роботи, що може створювати незручності для користувачів із вимогами до конфіденційності даних або тих, хто працює в умовах без доступу до стабільного інтернету.

1.3 Постановка задачі

Метою магістерської дисертації є створення інтелектуального інструменту, який автоматизує процес створення візуальних матеріалів на основі текстових описів, що сприятиме підвищенню ефективності та креативності у різних сферах, таких як маркетинг, наука та цифровий дизайн.

Призначення розробки – створення системи, здатної генерувати реалістичні та деталізовані зображення, що відповідають заданим текстовим описам, із підтримкою функцій налаштування деталізації та композиції.

До основних завдань, які необхідно виконати в магістерській дисертації, можна віднести наступні:

– Дослідити теоретичні аспекти використання генеративних моделей для генерації зображень, а також існуючі рішення.

– Систематизувати існуючі архітектури моделей перетворення текстових даних в зображення.

– Спроектувати систему генерації зображень з використанням моделі ймовірнісної дифузії.

– Розробити програмне забезпечення для системи генерації зображень з використанням моделі ймовірнісної дифузії.

– Провести тестування функціональних можливостей генерації зображень з використанням моделі ймовірнісної дифузії.

– Розробити стартап-проєкт з визначення потенційних стратегій виходу розробленої системи на ринок.

Висновку до розділу 1

У першому розділі розглянуто сучасні генеративні моделі для перетворення тексту в зображення, їхні архітектурні особливості, принципи роботи та можливості. Кожна з розглянутих моделей демонструє унікальні переваги та обмеження в контексті генерації візуального контенту на основі текстових описів. Imagen вирізняється здатністю генерувати високоякісні, деталізовані зображення завдяки каскадній дифузійній архітектурі, яка дозволяє поступово підвищувати роздільну здатність зображення. Stable Diffusion забезпечує гнучкість у відтворенні складних сцен і стилів.

Imagen, хоч і забезпечує виняткову якість та деталізацію, має обмежений доступ, оскільки інтегрована в закриту екосистему Google Gemini, що ускладнює її використання для незалежних розробників та комерційних проєктів. Stable Diffusion, попри свою гнучкість і відкритий доступ, може вимагати значних обчислювальних ресурсів для створення високоякісних зображень, а також потребує налаштування, що може бути складним для новачків. Платформа Runway ML є платною, і доступ до повного функціоналу можливий лише через підписку, що може бути фінансово обтяжливим для окремих користувачів або невеликих команд. Оскільки обчислення виконуються на серверній інфраструктурі, ефективність роботи платформи залежить від стабільного і швидкого інтернет-з’єднання, а у випадку технічних проблем чи високого навантаження на сервери можуть виникати затримки в обробці запитів. Runway ML не підтримує локальний режим роботи, що може створювати незручності для користувачів із вимогами до конфіденційності даних або тих, хто працює в умовах без доступу до стабільного інтернету.

2 ОГЛЯД ІСНУЮЧИХ АРХІТЕКТУР МОДЕЛЕЙ ПЕРЕТВОРЕННЯ ТЕКСТОВИХ ДАНИХ В ЗОБРАЖЕННЯ

2.1 Генеративні змагальні мережі

Генеративна змагальна мережа – це архітектура штучних нейронних мереж, розроблена для моделювання високовимірних розподілів даних через динамічне суперництво між двома нейронними мережами: генератором та дискримінатором. Ця концепція передбачає побудову системи, де генератор, навчаючись у неконтрольованому режимі, продукує зразки даних, які намагаються імітувати вихідний розподіл даних, тоді як дискримінатор оцінює автентичність цих зразків, приймаючи їх за реальні або штучні. Метою генератора є “обман” дискримінатора, що стимулює його виробляти дедалі реалістичніші зразки, адаптуючись до відгуків дискримінатора.

Дискримінатор, у свою чергу, постійно вдосконалює здатність розпізнавати згенеровані зразки, що створює самонавчальну систему зворотного зв’язку, яка моделює взаємний "еволюційний" процес. Ця взаємодія формує структуру гри з нульовою сумою, де прогрес одного агента зумовлює покращення стратегій іншого, наближаючи генератор до досягнення максимальної схожості зі справжніми даними. З математичної точки зору, навчання GAN полягає в оптимізації мінімакс-функції збитків, де генератор мінімізує ймовірність правильного класифікування дискримінатором, тоді як дискримінатор максимізує її [1].

В основі GAN лежить функція збитків (loss function), яка визначає цей процес і представлена як гра з нульовою сумою між двома моделями: генератором G та дискримінатором D.

Нехай $p_{data}(x)$ є справжнім розподілом даних, а $p_z(z)$ – розподілом латентного простору, з якого вибираються випадкові вектори z як вхідні значення для генератора $G(z;\theta_g)$, що генерує нові зразки. Дискримінатор $D(x;\theta_d)$ є класифікатором, який оцінює ймовірність того, що зразок x x є реальним, тобто

належить до $p_{data}(x)$, а не до $p_g(x)$, де $p_g(x)$ – розподіл, що генерується мережею G.

На рисунку 2.1 представлена типова архітектура генеративно-змагальної мережі.

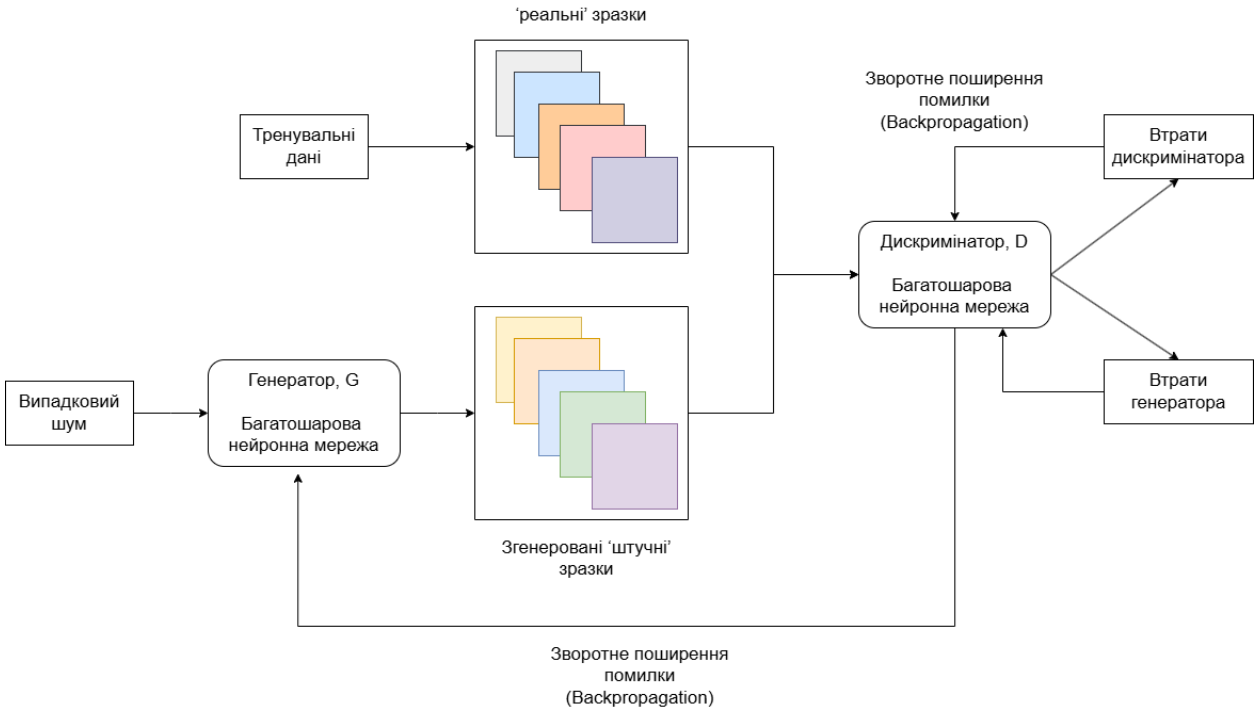

Рисунок 2.1 – Типова архітектура генеративно-змагальної мережі

Функція збитків для GAN визначається як (2.1):

$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \qquad (2.1)$$

З функції вище $D(x)$ максимізує функцію збитків, оскільки прагне правильно відрізняти справжні зразки від згенерованих.

Процес навчання є ітеративним і проводиться у два етапи:

– при фіксованому генераторі G дискримінатор D оновлює свої параметри θ_d, максимізуючи функцію збитків, тобто підвищуючи точність у класифікації реальних і підроблених даних.

− після оновлення дискримінатора генератор G оновлює свої параметри θ_g, мінімізуючи функцію збитків, зокрема, прагне обманути дискримінатор, змушуючи його прийняти згенеровані дані за справжні.

При ідеальному навчанні, коли G і D досягають Nash-рівноваги, дискримінатор не може відрізнити справжні дані від згенерованих, тобто $D(x) = \frac{1}{2}$ для всіх x. Це означає, що генеративний розподіл $p_g(x)$ збігається з справжнім розподілом даних $p_{data}(x)$, досягаючи умовної оптимізації.

Розглянемо одну з найвідоміших архітектур у сфері генеративних змагальних мереж – DCGAN (Deep Convolutional GAN), яка стала фундаментальною для генерації реалістичних зображень. Deep Convolutional Generative Adversarial Network (DCGAN) являє собою вдосконалену архітектуру генеративної змагальної мережі, яка інкорпорує глибинні згорткові нейронні мережі для забезпечення більш високої якості згенерованих зображень, ніж у класичних GAN. Основна ідея DCGAN полягає у застосуванні глибоких згорткових шарів у структурах генератора та дискримінатора, що значно покращує здатність моделі до відтворення складних розподілів і підвищує стабільність процесу навчання.

У DCGAN генератор побудований на базі транспонованих згорткових шарів, що забезпечують прогресивне масштабування вхідного латентного вектора до розмірів зображення. Вхідний латентний вектор z, обраний із розподілу $p(z)$, через послідовність транспонованих згорток поступово трансформується, що дозволяє синтезувати зображення з високим рівнем деталізації. Ключовою особливістю генератора є використання Batch Normalization у всіх шарах, окрім вихідного, що зменшує варіативність у активаціях та стабілізує процес навчання, дозволяючи уникнути проблеми вибуху або зникнення градієнтів.

Крім того, у всіх внутрішніх шарах генератора використовується функція активації ReLU, яка сприяє нелінійності моделі, тоді як вихідний шар використовує функцію активації $tanh$ для нормалізації діапазону значень пікселів у межах від -1 до 1. Дискримінатор у DCGAN є вдосконаленою глибокою згортковою мережею, аналогічною до класичної CNN, що застосовується для класифікації зображень. Він

приймає на вхід як реальні, так і згенеровані зразки, прагнучи визначити їхню автентичність. Використання шарів Batch Normalization також є критичним у дискримінаторі, адже воно стабілізує потік даних між шарами, що дозволяє значно зменшити зсуви активацій і забезпечує стабільніше оновлення параметрів під час навчання [4].

На рисунку 2.2 представлена архітектура генератору, який приймає вектор шуму z, і відображає його на виході G, який має розмір 64x64x3.

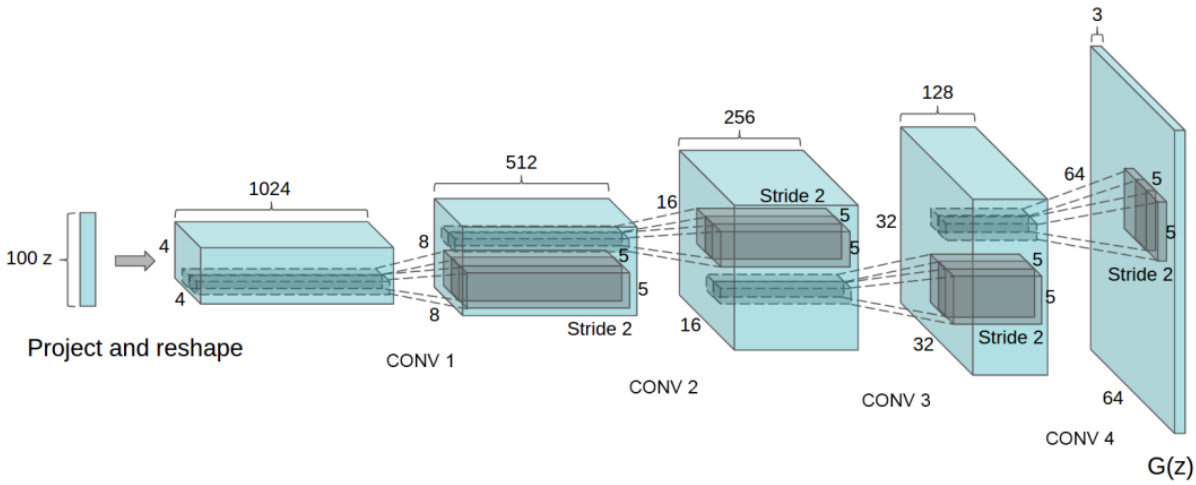

Рисунок 2.2 – Архітектура генератору DCGAN [6]

2.2 Варіаційні автокодувальники

Варіаційні автокодувальники (VAE, Variational Autoencoders) – це складні генеративні моделі, що базуються на байєсівській інтерпретації латентного простору і є вдосконаленням традиційних автокодувальників. Основна концепція VAE полягає у моделюванні апостеріорного розподілу прихованих змінних, які описують основні характеристики вхідних даних, шляхом максимізації відповідного нижнього оцінювача правдоподібності (ELBO, Evidence Lower Bound). Цей підхід дозволяє не лише кодувати вхідні дані в латентний простір, а й створювати нові зразки даних, що відповідають певному розподілу у латентному просторі, забезпечуючи можливість генерувати подібні до оригіналу дані.

Архітектура VAE складається з енкодера та декодера. Енкодер отримує вхідний зразок x x і перетворює його у параметри нормального розподілу (середнє μ та стандартне відхилення σ), що представляє приховану змінну z, яка визначає латентний простір.

Замість того, щоб кодувати вхід у фіксовану точку в латентному просторі, як це роблять класичні автокодувальники, VAE прагне навчити розподіл, у якому точки латентного простору слідують певному розподілу, наприклад, багатовимірному нормальному. Процес семплювання латентного простору в VAE здійснюється за допомогою так званого трикутникового трюку (reparameterization trick), де випадкова змінна z обчислюється за формулою $z = \mu + \sigma * \epsilon$, де ϵ – випадкова змінна, що вибирається з стандартного нормального розподілу $\mathcal{N}(0,1)$.

Функція втрат VAE складається з двох основних компонентів: відстані відновлення та регуляризаційного члена. Перша частина – це крос-ентропія між оригінальним і реконструйованим зразком, яка мінімізує різницю між вхідними даними x та їх реконструйованим аналогом x' із латентного простору. Друга частина – регуляризаційний член, що мінімізує розбіжність Кульбака-Лейблера (KL-дивергенцію) між розподілом латентного простору $q(z|x)$ та апостеріорним розподілом $p(z)$. KL-дивергенція стимулює латентний простір слідувати багатовимірному стандартному нормальному розподілу, що дозволяє моделі генерувати дані з добре структурованого латентного простору і уникати перенавчання на конкретні вхідні зразки [3].

2.3 Ймовірнісні моделі дифузії

Ймовірнісні моделі дифузії – це підхід у генеративному машинному навчанні, заснований на ідеї послідовного шумування та зворотного відновлення даних. Ці моделі використовують ітеративний процес, в якому спершу вводиться випадковий шум у дані, що поступово розмиває їхній вихідний розподіл, а потім відновлюють структуру, виконуючи зворотний дифузійний процес, що дозволяє отримати нові зразки, подібні до справжніх.

Основна ідея полягає в тому, щоб навчити модель поступово змінювати розподіл даних від складного до простого (зазвичай білого гаусівського шуму) шляхом накладення послідовних шарів випадкових перетворень. У прямому дифузійному процесі, який часто називають forward process, до даних додається шум, що призводить до поступового розмивання структури даних. У зворотному процесі (reverse process) модель навчається видаляти цей шум на кожному етапі, що дозволяє реконструювати дані з чистого шуму, генеруючи нові зразки, які відтворюють основні закономірності оригінального розподілу. Навчання ймовірнісних дифузійних моделей базується на мінімізації спеціальної функції втрат, що поєднує крос-ентропію між згенерованими та справжніми зразками і розбіжність Кульбака-Лейблера між послідовними станами зворотного процесу та відповідними станами прямого процесу. Моделі дифузії використовують стохастичний процес Маркова для опису послідовних переходів у forward та reverse процесах.

На рисунку 2.3 представлений процес прямої дифузії (forward diffusion):

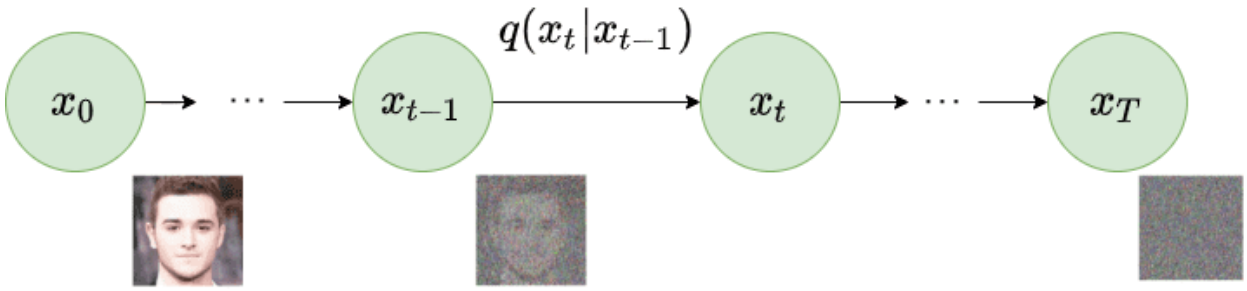

Рисунок 2.3 – Процес прямої дифузії [4]

У прямому процесі до вхідних даних x_0 поступово додається шум за допомогою Марківського ланцюга. Це призводить до того, що дані набувають дедалі більш випадкового вигляду з кожним кроком, аж поки вони не стають білим гаусівським шумом на кінцевому етапі. Прямий процес можна описати послідовністю станів $x_1, x_2, \dots, x_T$, де T – максимальна кількість кроків.

Процес задається наступним чином:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t;\ \mu_t = \sqrt{1-\beta_t}\, x_{t-1}, \sum_t \beta_t I\,, \tag{2.2}$$

де β_t – вказує на кожному кроці компроміс між інформацією, яку потрібно зберегти з попереднього кроку, та шумом, який необхідно додати, а I – одинична матриця. Цей параметр β_t зазвичай збільшується з кожним кроком, щоб рівномірно додавати шум протягом усього процесу дифузії.

Після кількох ітерацій x_T наближається до білого гаусівського шуму.

Його мета – поступово відновити структуру даних, видаляючи шум на кожному кроці. Зворотний процес описується умовною ймовірністю $p_\theta(x_{t-1}|x_t)$, яку моделюють нейронною мережею з параметрами θ.

Зворотній процес дифузії представлений на рисунку 2.4.

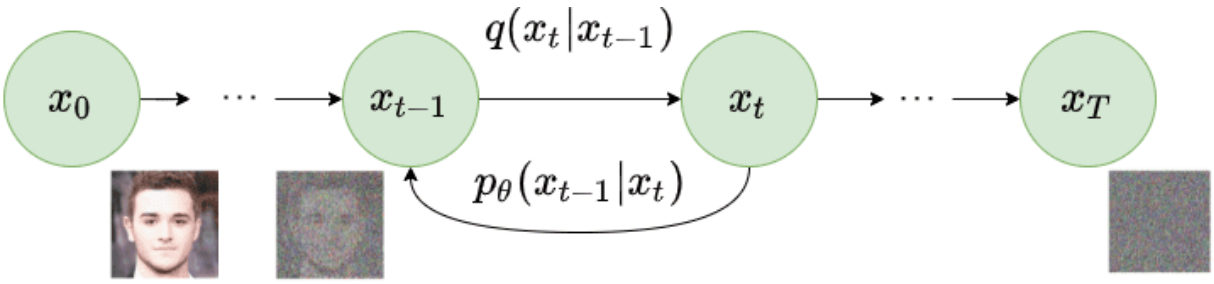

Рисунок 2.4 – Процес зворотньої дифузії [4]

Формально зворотний процес визначається як:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\big(x_{t-1}; \mu_0(x_t,t), \Sigma_\theta(x_t,t)\big), \tag{2.3}$$

де $\mu_0(x_t,t)$ та $\Sigma_\theta(x_t,t)$ – параметри середнього та коваріації, що прогнозуються на кожному кроці нейронною мережею.

Зворотний процес може бути переписаний у вигляді прогнозування шуму ϵ на кожному кроці t, де нейронна мережа прогнозує шум (2.4):

$$\epsilon_0(x_t,t) \approx \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\ \alpha_t}}, \tag{2.4}$$

де α_t – кумулятивний добуток 1 – β_t.

Це формулювання дозволяє моделі вивчати пряму регресійну задачу з прогнозування шуму, що значно спрощує оптимізацію та стабілізує навчання.

У дифузійних ймовірнісних моделях мета навчання полягає в тому, щоб наблизити зворотний розподіл $q(x_{t-1}|x_t)$ за допомогою параметризованого розподілу $p_\theta(x_{t-1}|x_t)$. Оскільки зворотний розподіл є невідомим і залежить від складного апостеріорного розподілу даних, його апроксимація є центральною задачею навчання моделі. Для оцінки якості апроксимації використовується варіаційна нижня межа (VLB), яка є нижньою межею логарифмічної правдоподібності генерації даних. Ця межа визначається як:

$$\log_{p_\theta}(x_0) \geq -\mathcal{L}, \tag{2.5}$$

де функція втрат $\mathcal{L}$ виражається через математичне очікування (2.6):

$$\mathcal{L} = \mathbb{E}_q\left[-\log_{p_\theta}(x_0) + \sum_{t=1}^{T} D_{KL}\big(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)\big)\right], \tag{2.6}$$

де $-\log_{p_\theta}(x_0)$ – правдоподібність початкових даних x_0, яка оцінює, наскільки добре модель відновлює початковий розподіл після проходження зворотного процесу; $D_{KL}\big(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)\big)$ – відповідає за мінімізацію відстані між розподілом q і апроксимацією p_θ для кроку t.

Далі, розглянемо використання процесів дифузії в латентному просторі. Латентна дифузійна моделі [10] виконує процес дифузії у латентному просторі замість піксельного, що знижує витрати на навчання та прискорює процес генерації. Цей підхід базується на спостереженні, що більшість інформації в зображенні відповідає за його візуальні деталі, тоді як семантична та концептуальна структура зберігається навіть після сильного стиснення.

На рисунку 2.5 представлено графік компромісу між рівнем стиснення та спотворенням, який ілюструє два етапи стиснення: сприйнятливе (perceptual) та семантичне (semantic) стиснення.

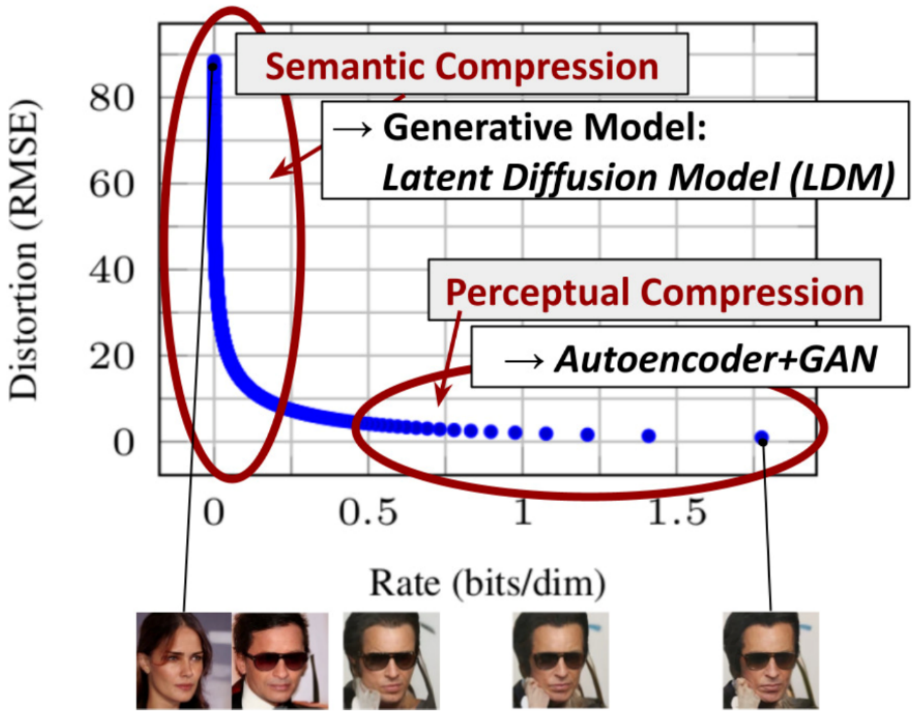

Рисунок 2.5 – Графік компромісу між рівнем стиснення та спотворенням, який ілюструє два етапи стиснення: сприйнятливе та семантичне стиснення [11]

Процес сприйнятливого стиснення базується на моделі автоенкодера. Енкодер використовується для стиснення вхідного зображення $x \in \mathbb{R}^{H*W*3}$ у менший двовимірний латентний вектор $z = \varepsilon(x) * \mathbb{R}^{h*w*c}$, де коефіцієнт зменшення розмірності становить $f = H/h = W/w = 2^m, m \in \mathbb{N}$.

Потім декодер реконструює зображення з латентного вектора $\tilde{x} = D(z)$.

У статті [11] досліджуються два типи регуляризації під час навчання автоенкодера для уникнення занадто високої дисперсії у латентних просторах:

– Невелике покарання KL-дивергенцією щодо стандартного нормального розподілу в навченому латентному просторі, подібно до VAE.

– Використання шару векторного квантування у межах декодера, аналогічно до VQVAE, але шар квантування інтегрується в декодер.

Процеси дифузії та денойзингу відбуваються в латентному векторі z. Модель денойзингу являє собою U-Net, умовлений часом, розширений механізмом крос-уваги (cross-attention) для роботи з гнучкою умовною інформацією для генерації зображень (наприклад, класові мітки та семантичні карти). Такий дизайн дозволяє об'єднувати представлення у модель за допомогою механізму крос-уваги. Кожен тип умовної інформації поєднується з доменним енкодером τ_θ, який проектує умовний вхід y у проміжне представлення, яке можна інтегрувати у компонент крос-уваги, $\tau_\theta \in \mathbb{R}^{M*d_\tau}$:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.7}$$

де $Q = W_Q^{(i)} * \varphi_i(z_i)$, $K = W_K^{(i)} * \tau_\theta(y)$, $V = W_V^{(i)} * \tau_\theta(y)$.

В дифузійних моделях для витягування ознак з вхідних даних зазвичай використовуються дві популярні архітектури: U-Net та Transformer.

U-Net включає два основних компоненти: блок зменшення розмірності (downsampling) і блок збільшення розмірності (upsampling):

– Зменшення розмірності: Кожен етап складається з двох послідовних згорткових операцій 3x3 без заповнення (unpadded convolutions), після кожної з яких застосовується активація ReLU і операція 2x2 max-pooling зі зсувом 2.

– Збільшення розмірності: На кожному етапі виконується масштабування карти ознак, за яким йде згортка 2x2, що зменшує кількість каналів функцій у два рази.

– Зв'язки між рівнями: Прямі з'єднання між відповідними шарами блоків зменшення та збільшення розмірності забезпечують передачу високороздільних ознак, необхідних для ефективної реконструкції даних. Це реалізується шляхом об'єднання відповідних шарів з обох блоків.

На рисунку 2.6 представлено архітектуру U-Net, яка зазвичай використовується в дифузійних моделях для генерації зображень.

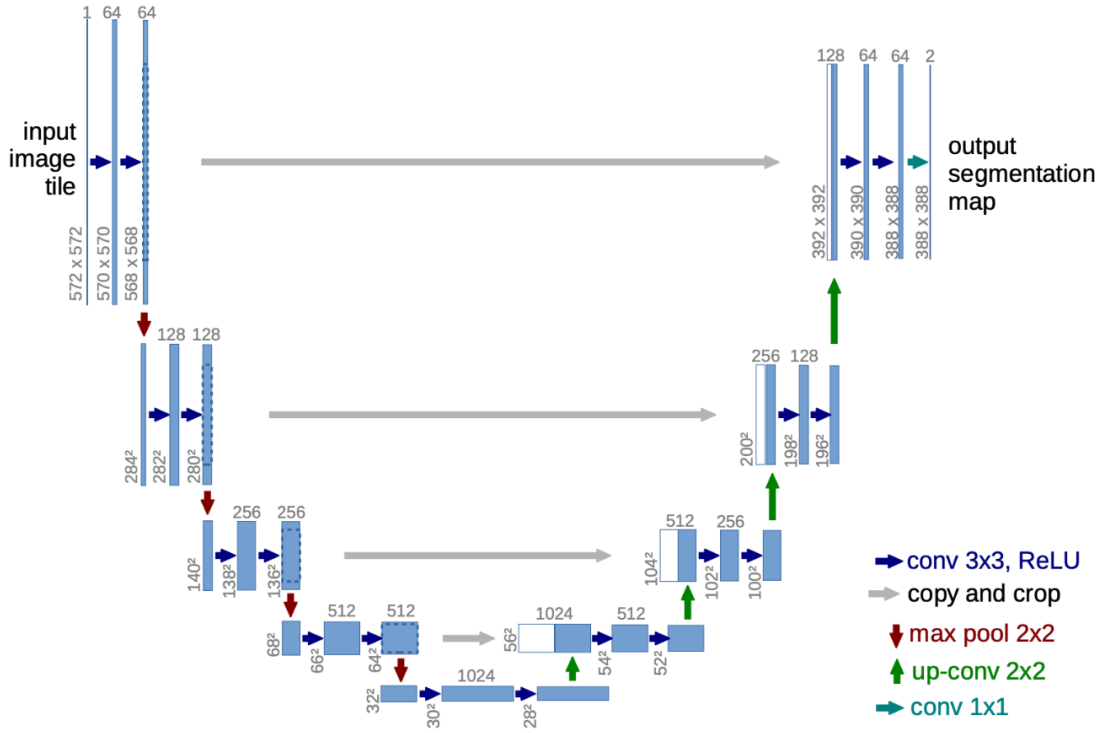

Рисунок 2.6 – Архітектура U-Net, яка використовується в дифузійних моделях [12]

Для забезпечення генерації зображень з урахуванням додаткових даних, таких як контури Кенні, ескізи користувача, скелети людських поз, карти сегментації, глибини та нормалі, ControlNet [13] вводить зміни в архітектуру шляхом додавання «вкладених» нульових згорток у вигляді тренованої копії початкових ваг кожного шару енкодера в U-Net.

Зокрема, для нейронного блоку $\mathcal{F}_\theta$, ControlNet виконує наступне:

– Заморожує початкові параметри θ оригінального блоку $\mathcal{F}_\theta$.

– Створює копію блоку з тренованими параметрами θ_c і додає додатковий умовний вектор c.

– Використовує два шари нульових згорток, позначені як $\mathcal{Z}_{\theta_{z1}}$ та $\mathcal{Z}_{\theta_{z2}}$. Це шари 1x1 згорток, де початкові ваги та зсуви встановлені в нулі. Нульові згортки запобігають внесенню випадкових шумів у градієнти на ранніх етапах.

Фінальний вихід визначається як (2.8):

$$y_c = \mathcal{F}_\theta(x) + \mathcal{Z}_{\theta_{z2}}\left(\mathcal{F}_{\theta_c}\left(x + \mathcal{Z}_{\theta_{z1}}(c)\right)\right). \quad (2.8)$$

Diffusion Transformer, який досліджено в [15], розроблений для моделювання дифузії, працює з латентними патчами, використовуючи той самий принцип, що й Latent Diffusion Model.

Архітектура DiT має наступні особливості:

– Латентне представлення вхідних даних z подається як вхід у дифузійний трансформер.

– Латентний шум $z_{\text{шум}}$ розбивається на патчі розміром $p * p$ та перетворюється на послідовність токенів розміром $n * d$, де n – кількість патчів, а d – розмірність ознак кожного патчу.

– Ця послідовність токенів проходить через трансформерні блоки. Для умовної генерації на основі додаткової інформації, такої як часовий крок t або класова мітка y, було досліджено три підходи [15, 16]. Найкращі результати продемонстрував підхід adaLN (адаптивна нормалізація шарів) із нульовими початковими значеннями (Zero). Він перевершив методи умовного навчання в контексті та використання блоків крос-уваги.

– У цьому підході параметри масштабування та зсуву (γ, β) розраховуються як функція від суми векторів ембеддінгів t і y.

Окрім цього, параметри масштабування λ визначаються окремо для кожного виміру та застосовуються перед будь-якими залишковими з'єднаннями всередині блоків.

Для створення зображень високої якості та роздільної здатності, в роботі [16] запропонували використовувати послідовність кількох дифузійних моделей, кожна з яких працює з поступово зростаючою роздільною здатністю. Важливим компонентом цієї архітектури є шумова умовна аугментація, яка виконується між моделями в цьому конвеєрі. Суть полягає у сильній аугментації даних, що використовуються як умовний вхід z для кожної моделі суперрезолюції $p_\theta(x|z)$. Додавання умовного шуму допомагає знизити накопичення помилок у

послідовності моделей. Для генерації зображень високої роздільної здатності U-Net є поширеним вибором архітектури.

На рисунку 2.7 представлений приклад використання послідовності декількох дифузійних моделей.

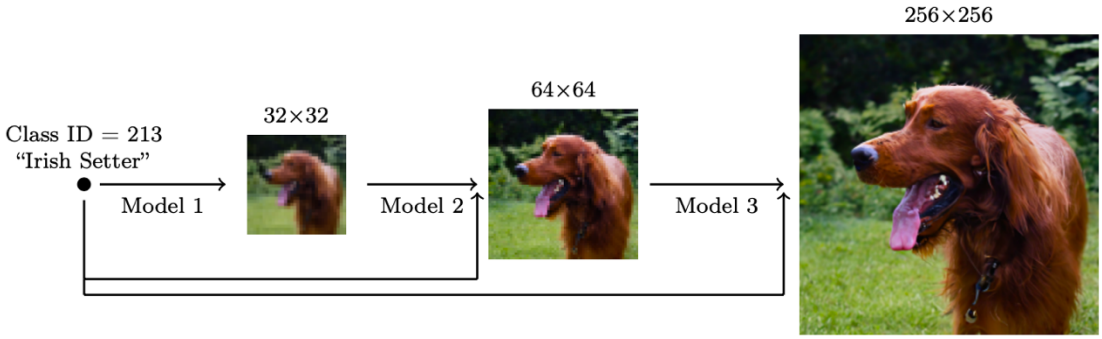

Рисунок 2.7 – Концепція використання послідовності декількох дифузійних моделей [16]

Найефективнішим шумом виявилося додавання гаусівського шуму на низьких роздільних здатностях та гаусівського розмиття на високих. Окрім цього, були досліджені два підходи до аугментації умовних даних, які вимагають незначних змін у процесі навчання. Умовний шум додається лише на етапі навчання і не використовується під час генерації.

Дворівнева дифузійна модель unCLIP [17] значною мірою використовує текстовий енкодер CLIP для створення високоякісних зображень на основі тексту.

Використовуючи попередньо натреновану модель CLIP та набір даних, що містить пари (x, y), де x – це зображення, а y – відповідний текстовий опис, обчислюються текстові та візуальні ембеддинги CLIP, позначені як $c^t(y)$ і $c^i(x)$ відповідно. Модель unCLIP одночасно навчає два компоненти:

– Попередню модель $P(c^i|y)$ – генерує ембеддинг зображення c^i на основі текстових даних y.

– Декодер $P(x|c^i, [y])$ – створює зображення x використовуючи CLIP ембеддинг зображення c^i і, за потреби, оригінальний текст y.

Ці дві моделі забезпечують умовну генерацію, оскільки (2.9):

$$\underbrace{P(x|y) = P\left(x, c^i|y\right)} = P\left(x, c^i|y\right)P\left(c^i|y\right). \quad (2.9)$$

Отже, unCLIP реалізує двоетапний процес генерації зображень:

– На основі тексту t модель CLIP створює текстовий ембеддинг $c^t(y)$. Використання латентного простору CLIP дозволяє виконувати маніпуляції із зображеннями на основі тексту без додаткового навчання.

– Дифузійна або автогрегресивна модель $P\left(c^i|y\right)$ обробляє текстовий ембеддинг CLIP, формуючи ембеддинг зображення, який потім використовується дифузійним декодером $P\left(x|c^i, [y]\right)$ для створення зображення. Декодер також може генерувати варіації існуючих зображень, зберігаючи їхній стиль та семантичну структуру.

Замість моделі CLIP, Imagen [18] використовує попередньо навчену велику мовну модель (заморожений текстовий енкодер T5-XXL) для кодування тексту під час генерації зображень. Загальна тенденція полягає в тому, що збільшення розміру моделі покращує якість зображень і відповідність між текстом та зображеннями. Дослідження [19, 20] показали, що T5-XXL і текстовий енкодер CLIP демонструють схожу продуктивність на MS-COCO, однак під час оцінки людиною на наборі тестових запитів DrawBench (що включає 11 категорій) перевага віддається T5-XXL. Під час використання «classifier-free guidance» збільшення коефіцієнта ω може покращити відповідність між текстом і зображенням, але водночас погіршує візуальну якість зображення. Це викликано невідповідністю між даними під час навчання та тестування: якщо під час навчання значення ω обмежувалось діапазоном $[a, b]$, то під час тестування також слід дотримуватись цього обмеження. Для цього запропоновано два методи обмеження:

– Статичне обмеження: обмеження прогнозованих значень ω до діапазону $[a, b]$.

– Динамічне обмеження: на кожному кроці семплінгу обчислюється певний процентиль s абсолютних значень пікселів; Якщо $s > 1$, прогноз обмежується до $[-s, s]$ та ділиться на s.

2.4 Авторегресивні моделі

Авторегресивні моделі є одним із фундаментальних підходів у генеративному машинному навчанні, де складні багатовимірні розподіли моделюються через послідовний розрахунок умовних ймовірностей.

PixelCNN і PixelRNN – це авторегресивні моделі, що моделюють спільний розподіл пікселів у зображенні, використовуючи автокореляційну залежність між ними.

Формально, вони розкладають спільний розподіл пікселів $x = \{x_1, x_2, \dots, x_N\}$ на послідовність умовних ймовірностей відповідно до авторегресивного принципу (2.10):

$$p(x) = \prod_{i=1}^{N} p(x_i | x_1, x_2, \dots, x_{i-1}), \tag{2.10}$$

де x_i – інтенсивність i-го пікселя в певному колірному каналі або в градаціях сірого, N – кількість пікселів у зображенні. Цей підхід дозволяє моделі поетапно генерувати зображення, обчислюючи значення кожного пікселя x_i на основі значень попередніх $x_{1:i-1}$.

У PixelCNN, умовна ймовірність $p(x_i | x_{1:i-1})$ моделюється за допомогою згорткових операцій. Для забезпечення авторегресивності в PixelCNN використовуються масковані згорткові фільтри, які блокують доступ до майбутніх пікселів. Маска розділяє фільтр на дві частини: одна охоплює попередні пікселі, а інша – майбутні.

Формально, активації шару визначаються як (2.11):

$$h_l(x) = \sigma(W_l * h_{l-1} + b_l), \tag{2.11}$$

де $h_l(x)$ – активація l-го шару, W_l – параметри маскованого фільтра, b_l – зсув, а σ – нелінійна функція активації, наприклад, ReLU.

Архітектура PixelCNN має обмеження на обробку глобального контексту через фіксований розмір рецептивного поля, що робить її менш ефективною для складних залежностей між пікселями.

У PixelRNN використовуються рекурентні мережі для захоплення довготривалих залежностей між пікселями. У цій архітектурі кожен рядок пікселів обробляється рекурентно вздовж горизонтального та вертикального напрямків, дозволяючи враховувати контекст не тільки від попередніх пікселів у поточному рядку, але й від попередніх рядків.

Формально, активації в PixelRNN визначаються як (2.12):

$$h_l(x_i) = \phi(W_l h_{l-1}(x_{1:i-1}) + U_l h_l(x_{1:i-1}) + b_l), \tag{2.12}$$

де W_l та U_l – ваги згорток для вертикальної та горизонтальної залежностей відповідно, а ϕ – активаційна функція (наприклад, LSTM або GRU). Цей підхід дає можливість обробляти більший контекст за рахунок послідовного обчислення, але він є обчислювально більш витратним у порівнянні з PixelCNN.

Обидві моделі, PixelCNN і PixelRNN, підтримують багатоканальну генерацію (RGB) шляхом розбиття умовної ймовірності на канали (2.13):

$$p(x_i) = p\left(x_i^R | x_{1:i-1}\right)\left(x_i^G | x_{1:i-1}, x_i^R\right)\left(x_i^B | x_{1:i-1}, x_i^R, x_i^G\right), \tag{2.13}$$

де x_i^R, x_i^G, x_i^B – значення відповідних каналів кольору. Це дозволяє моделі враховувати залежності між кольоровими каналами під час генерації.

PixelCNN і PixelRNN також мають суттєві особливості в реалізації, які впливають на їхню продуктивність та область застосування. Це дозволяє моделі навчитися зв’язку між умовними ознаками та структурами зображення,

забезпечуючи генерацію зображень відповідно до заданих умов. Ключовою інновацією є використання механізмів шлюзів (gating mechanisms) у розширених версіях моделей, таких як Gated PixelCNN. Ця модифікація додає структуровані шари, які дозволяють моделі адаптивно контролювати потік інформації через шари мережі, що знижує ризик перенасичення ознаками та покращує якість генерації. Наприклад, шлюзи дозволяють враховувати різні рівні деталей – від глобального контексту до локальних текстур, що важливо для точного відтворення складних візуальних структур.

На рисунку 2.8 представлено шар в архітектурі Gated PixelCNN.

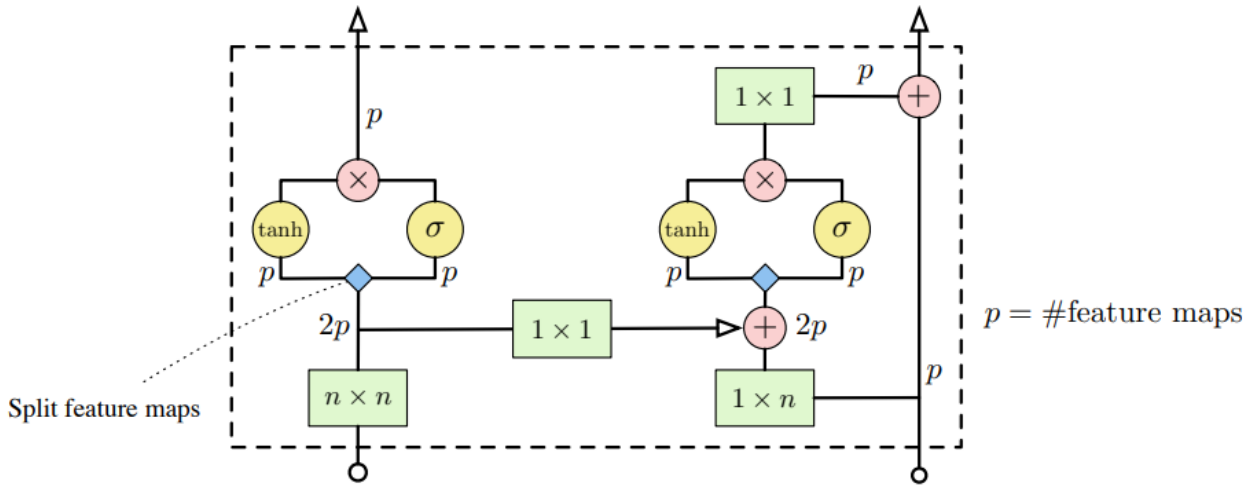

Рисунок 2.8 – Один шар в архітектурі Gated PixelCNN [21]

Щодо обчислювальної складності, оптимізація архітектури PixelRNN дозволила зменшити затримку, пов’язану з рекурентними обчисленнями. Наприклад, використання горизонтальних і вертикальних рекурентних потоків у Parallelized PixelRNN дозволяє обробляти інформацію більш ефективно, значно скорочуючи час навчання та генерації. Однак навіть із цими оптимізаціями PixelRNN залишається більш ресурсоємною, ніж PixelCNN. Ще однією перспективною областю є інтеграція PixelCNN та PixelRNN у мультимодальні системи. Наприклад, у завданнях, де зображення генеруються на основі текстових описів, ці моделі можуть використовувати текстові вектори як додаткову умову, що дозволяє створювати зображення із заданими характеристиками. Подібні системи

розширюють межі можливостей генеративних моделей, дозволяючи працювати із завданнями, що потребують високої точності в збереженні як семантичної, так і візуальної відповідності.

2.5 Моделі на базі нормуючих потоків

Normalizing Flows (нормуючі потоки) – це клас генеративних моделей, які дозволяють трансформувати простий базовий розподіл, наприклад, багатовимірний гаусівський, у складний розподіл, що відповідає структурі зображень. Це досягається через послідовність оборотних і диференційованих перетворень. Основна ідея полягає в тому, щоб за допомогою цих перетворень навчити модель з високою точністю відтворювати реальний розподіл пікселів, що спостерігається в зображеннях. Мета нормуючих потоків – у виконанні послідовних трансформацій даних. Початковий вхід z_0 вибирається з простого розподілу, наприклад, стандартного гаусівського $p_0(z_0)$.

Після цього до нього застосовується серія трансформацій $f_1, f_2, \dots, f_K$, які поступово модифікують його розподіл. У результаті отримується складний розподіл z_K, що відповідає структурі зображення.

Щільність у новому просторі z_K можна обчислити за допомогою правила зміни змінної (2.14):

$$p_K(z_K) = p_0(z_0) \prod_{i=1}^{K} \left| det \frac{\partial f_i}{\partial z_{i-1}} \right|^{-1}, \tag{2.14}$$

де $\frac{\partial f_i}{\partial z_{i-1}}$ – якобіан перетворення f_i, а $\left| det \frac{\partial f_i}{\partial z_{i-1}} \right|$ – його визначник.

Якобіан відіграє важливу роль, оскільки він визначає, як масштабуються ймовірності під час трансформації.

Вибір трансформацій f_i є критично важливим і має задовольняти три умови:

– Кожна трансформація повинна мати обернену функцію f_i^{-1}, що дозволяє переходити між просторами.

– Щоб забезпечити обчислюваність якобіана.

– Визначник якобіана має бути простим для обчислення.

Одним із популярних методів є використання афінних шарів (Affine Coupling Layers). У таких шарах змінні розбиваються на дві частини: одна залишається незмінною, а інша трансформується через афінне перетворення, параметризоване нейронною мережею. Це дозволяє зберігати простоту обчислень, оскільки якобіан такого перетворення є діагональною матрицею. Для генерації зображень нормуючі потоки демонструють високі результати завдяки своїй здатності враховувати складні просторові залежності між пікселями. Наприклад, у моделі Glow використовуються послідовності афінних шарів, що дають змогу ефективно кодувати й декодувати зображення.

Розглянемо Continuous Normalizing Flows і Masked Autoregressive Flow, які є двома різними підходами в рамках нормуючих потоків, кожен із яких пропонує специфічні властивості для моделювання складних розподілів, таких як зображення. Continuous Normalizing Flows базуються на ідеї моделювання трансформації даних через безперервний потік.

На відміну від традиційних нормуючих потоків, які використовують дискретну послідовність перетворень, CNF моделює цей процес як розв’язок диференціального рівняння, яке описує зміну розподілу даних у часі.

Формально, якщо $z(t)$ представляє стан даних у момент часу t, то його еволюцію можна описати рівнянням (2.15):

$$\frac{dz(t)}{dt} = f(z(t), t; \theta), \tag{2.15}$$

де f – це параметризована функція (зазвичай нейронна мережа), яка визначає динаміку потоку, а θ – її параметри.

Початковий стан $z(0)$ береться з простого базового розподілу (наприклад, гаусівського), а фінальний стан $z(T)$ відповідає розподілу даних.

Важливою перевагою CNF є можливість працювати з безперервними трансформаціями, що забезпечує високу гнучкість у моделюванні складних залежностей.

В обчислювальному плані обчислення щільності виконується через формулу (2.16):

$$\log p\big(z(T)\big) = \log p\big(z(0)\big) - \int_0^T Tr\left(\frac{\partial f(z(t),t)}{\partial z}\right) dt\,, \tag{2.16}$$

де Tr – слід матриці якобіана.

CNF відомі своєю здатністю точно моделювати високовимірні розподіли, але вони є обчислювально інтенсивними, оскільки потребують чисельного інтегрування, яке включає якобіан на кожному кроці.

Masked Autoregressive Flow використовує інший підхід, поєднуючи ідеї нормуючих потоків і авторегресивного моделювання. Основна ідея MAF полягає у використанні авторегресивних залежностей для побудови перетворення між простим розподілом і складним. У MAF перетворення даних відбувається шляхом застосування авторегресивних функцій, які забезпечують маскування залежностей між змінними. Для кожної змінної z_i, її перетворення залежить лише від попередніх змінних $\{z_1, z_2, \dots, z_{i-1}\}$:

$$z_i' = \mu_i(z_{1:i-1}) + \sigma_i(z_{1:i-1}) * z_i, \tag{2.17}$$

де μ_i і σ_i – функції з параметрами, що відповідають зсуву і масштабуванню, які обчислюються нейронними мережами.

Головною особливістю MAF є те, що обчислення визначника якобіана значно спрощується, оскільки матриця якобіана є трикутною, а її визначник є просто добутком діагональних елементів. Це дозволяє ефективно обчислювати щільність і робить MAF придатним для великих датасетів.

MAF є потужним інструментом для моделювання складних розподілів, але вимагає послідовного обчислення залежностей, що може сповільнювати генерацію даних. На практиці MAF часто використовується для задач, де пріоритетом є точність оцінки щільності, а не швидкість генерації.

Незважаючи на успіх нормуючих потоків у моделюванні високовимірних щільностей, їхні архітектури мають певні недоліки. Перш за все, їхній латентний простір, у який проектуються вхідні дані, не є простором із меншою кількістю вимірів. Це означає, що моделі на основі потоків за замовчуванням не забезпечують стиснення даних і вимагають значних обчислювальних ресурсів. Проте, існують методи, які дозволяють використовувати ці моделі для компресії зображень. Ще одним важливим обмеженням моделей нормуючих потоків є їхня слабкість у правильному оцінюванні ймовірності для вибірок, що знаходяться поза розподілом (out-of-distribution), тобто таких, які не відповідають розподілу навчальних даних. Для пояснення цього явища висунуто кілька гіпотез.

Серед них гіпотеза про «типовий набір», яка стверджує, що такі моделі можуть не захоплювати всі властивості даних через обмеження у виборі «типових» прикладів; проблеми з оцінкою параметрів під час навчання; або ж фундаментальні проблеми, пов’язані з ентропією розподілу даних. Однією з найцікавіших властивостей нормуючих потоків є оборотність (invertibility) їхніх бієктивних відображень, які моделі вчаться будувати. Ця властивість забезпечується конструктивними обмеженнями в архітектурі моделей, такими як у RealNVP або Glow, що теоретично гарантують можливість оберненого перетворення. Оборотність має критичне значення для застосування теореми про зміну змінних, обчислення якобіана відображення, а також для семплінгу нових даних. Проте на практиці ця властивість може порушуватися через чисельну неточність, що призводить до нестабільності й «вибуху» зворотного перетворення. Це явище значною мірою обмежує практичну застосовність нормуючих потоків у завданнях, які потребують точності в моделюванні [24].

Висновки до розділу 2

У другому розділі представлено огляд основних архітектур генеративних моделей для перетворення текстових даних у зображення, кожна з яких має свої унікальні переваги, обмеження та сфери застосування.

Генеративні змагальні мережі виділяються своєю здатністю синтезувати реалістичні зображення через динамічну взаємодію генератора та дискримінатора, де класичні підходи стали основою для створення складних візуальних структур.

Варіаційні автокодувальники акцентуються на моделюванні латентного простору, що дозволяє не лише відновлювати, але й створювати нові дані зі збереженням ключових характеристик.

Ймовірнісні моделі дифузії демонструють високу реалістичність шляхом поступового відновлення зображень із шуму, використовуючи складні ймовірнісні процеси, які можуть бути реалізовані через ефективні архітектури, такі як латентна дифузійна модель.

Авторегресивні моделі показують себе добре у завданнях генерації послідовностей пікселів, хоча їхня обчислювальна складність залишається викликом.

Нормуючі потоки демонструють здатність трансформувати прості розподіли у складні, але через високі обчислювальні вимоги та певні обмеження в моделюванні залишаються менш поширеними.

Таким чином, проведений огляд показав, що найкращою для реалізації у розроблюваній системі буде ймовірнісна модель дифузії, тому що вона забезпечує високу реалістичність згенерованих зображень завдяки поступовому процесу відновлення сигналу із шуму. Крім того, дифузійна модель демонструє високу гнучкість у роботі зі складними текстовими описами, адаптуючись до різних стилістичних та композиційних вимог. Її архітектурна ефективність, особливо у вигляді латентної дифузійної моделі, знижує обчислювальну складність порівняно з прямим моделюванням пікселів, що робить її придатною для масштабованих систем із обмеженими ресурсами.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Генерація зображень

На схемі, яка представлена в додатку Б, представлено алгоритм знешумлення зображень з використанням моделі ймовірісної дифузії.

На схемі зображено алгоритм роботи денойзера, який є центральним компонентом дифузійної моделі. Цей процес базується на поступовому зменшенні шуму в зашумленому зображенні x_t і відновленні його до менш зашумленого стану x_{t-1} що наближає зображення до його оригінального вигляду. На вхід алгоритму подається зашумлене зображення x_tяке є результатом багаторазового додавання шуму до початкового зображення. Відновлення зображення здійснюється через комбінацію кількох важливих компонентів. Першим етапом є використання кодувальника часових кроків, який отримує поточний часовий індекс t що вказує на стадію процесу дифузії. Цей кодувальник формує часовий ембеддинг emb_t який інформує модель про те, як саме шум потрібно видаляти на цьому кроці. Якщо модель є умовною, тобто використовує додатковий контекст, текстовий кодувальник отримує текстовий опис, наприклад, "Пустельний оазис із невеликим ставком, оточений пальмами під нічним зоряним небом". Цей текст перетворюється на багатовимірний вектор, який передається в мережу разом із часовим ембеддингом. Ці два елементи – часова і текстова інформація – є важливими, оскільки вони забезпечують умови для правильного відновлення зображення.

Центральним компонентом є мережа прогнозування шуму $\epsilon_\theta(x_t, t)$. Вона отримує вхідне зашумлене зображення x_t, часовий ембеддинг emb_t, а також, за необхідності, текстовий контекст. Завдання мережі – передбачити шум, який було додано до сигналу на цьому кроці, тобто визначити, яку частину x_t становить шум, а яку – корисний сигнал. Цей прогноз використовується для подальшого зменшення шуму. Спершу, передбачений шум ϵ_θ масштабується коефіцієнтом $\frac{1-a_t}{\sqrt{1-\overline{a_t}}}$, що дозволяє точно скоригувати шумовий компонент. Цей шумовий

компонент потім віднімається від x_t, щоб залишити сигнал із меншим рівнем шуму. Однак, для забезпечення стабільності процесу і уникнення надмірного "перечищення" сигналу, додається невелика кількість нового шуму z, який генерується із стандартного нормального розподілу $\mathcal{N}(0, I)$. Цей новий шум масштабується параметром σ_t, що відповідає за правильну варіацію результату.

У результаті цих операцій отримується знешумлене зображення x_{t-1}, яке є проміжним станом між початковим зашумленим x_t та чистим x_0.

Таким чином, на кожному кроці алгоритм виконує прогнозування шуму, коригування сигналу та додавання регульованого нового шуму, поступово відновлюючи зображення до початкового стану. Уся ця процедура є основою для роботи дифузійних моделей, які демонструють високу ефективність у генерації реалістичних зображень в умовному та в безумовному середовищах.

3.2 Метрики оцінювання якості

Оцінювання якості генеративних моделей, зокрема дифузійних моделей (наприклад, DDPM), вимагає використання метрик, які здатні кількісно визначати, наскільки добре згенеровані зображення відповідають розподілу реальних зображень. Серед найпоширеніших метрик, які використовуються для оцінювання таких моделей, є Inception Score та Fréchet Inception Distance.

Inception Score (IS) є однією з перших метрик, що широко використовувалась для оцінювання генеративних моделей. Вона вимірює два основні аспекти якості зображень: їхню інформативність (різноманіття класифікаційних ймовірностей) і різноманіття (відмінність між зображеннями). Концептуально Inception Score базується на передбаченні класифікатора $p(y|x)$, де y – це клас, а x – згенероване зображення. Для моделі вважається бажаним, щоб кожне зображення чітко належало до одного класу (низька ентропія $H(y|x)$), і при цьому розподіл класів $p(y)$ у всьому наборі згенерованих зображень був максимально рівномірним (висока ентропія $H(y|x)$). Формально IS визначається як експонента від середнього значення узагальненої дивергенції Кульбака-Лейблера між $p(y|x)$ та $p(y)$ (3.1):

$$IS = xp\left(\mathbb{E}_{x\sim p_{data}}\left[KL\left(p(y|x)||p(y)\right)\right]\right), \quad (3.1)$$

де $p(y) = \int p(y|x)p_{data}(x)\,dx$ – це середній розподіл класів у згенерованих зображеннях. Чим вище значення IS, тим краща якість зображень. Однак, IS має кілька обмежень: вона не враховує відповідність згенерованих зображень реальному розподілу і є чутливою до вибору класифікатора. На противагу цьому, Fréchet Inception Distance є більш сучасною метрикою, яка порівнює реальні та згенеровані зображення, використовуючи простір характеристик, екстрагованих з останнього прихованого шару моделі Inception. Основна ідея FID полягає в тому, що реальні та згенеровані зображення повинні мати схожі розподіли в просторі ознак. Вважається, що ці ознаки представляють високоінформативні характеристики зображень, що робить метрику стійкою до низькорівневих змін, таких як шум. Розподіли характеристик $\mathcal{N}(\mu_r, \Sigma_r)$ для реальних зображень і $\mathcal{N}\left(\mu_g, \Sigma_g\right)$ для згенерованих описуються багатовимірними нормальними розподілами з середніми значеннями μ_r, μ_g та коваріаційними матрицями Σ_r, Σ_g. FID обчислюється як відстань між цими двома розподілами:

$$FID = \left\|\mu_r - \mu_g\right\|_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2 * \sqrt{\Sigma_r * \Sigma_g}\right), \quad (3.2)$$

де $\left\|\mu_r - \mu_g\right\|_2^2$ – це квадратична відстань між середніми векторами, яка вимірює зсув розподілів, а $Tr\left(\Sigma_r + \Sigma_g - 2 * \sqrt{\Sigma_r * \Sigma_g}\right)$ – враховує різницю в коваріаціях, визначаючи подібність у формі розподілів. FID має низку переваг, зокрема врахування якості зображень, так і їхнього розподілу, і є більш стійким до шуму, але вимагає великого обсягу даних для точного оцінювання. Чим менше значення FID, тим краще згенеровані зображення відповідають реальним. Таким чином, IS і FID забезпечують комплексну оцінку якості генеративних моделей, хоча кожна з них має свої переваги й обмеження. IS підкреслює індивідуальну

якість зображень та їхню різноманітність, тоді як FID більш точно оцінює відповідність розподілів.

Висновки до розділу 3

У третьому розділі деталізовано математичне забезпечення, зокрема алгоритм роботи дифузійної моделі, який базується на поступовому відновленні сигналу з урахуванням часових і текстових умов, використовуючи передбачення шуму через обчислювальні графи та параметричні функції.

Проаналізовано ключові метрики оцінювання якості згенерованих зображень, які враховують інформативність та відповідність розподілу даних.

4 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ

4.1 Огляд використаних технічних засобів

Для розробки системи генерації зображень з використанням ймовірісної моделі дифузії було обрано наступні технічні засоби: мову програмування Python, а також бібліотеки Torch, TorchVision, diffusers та clip.

4.1.1 Мова програмування Python

Python – це високорівнева інтерпретована мова програмування з динамічною типізацією та суворим управлінням пам’яттю, яка забезпечує потужну інфраструктуру для реалізації задач машинного навчання та глибинного навчання. Вона використовує віртуальну машину CPython для виконання коду та надає підтримку численних вбудованих бібліотек для лінійної алгебри, обчислень із багатовимірними масивами і аналізу даних, таких як NumPy, а також високорівневих абстракцій для роботи з нейронними мережами, що реалізуються через TensorFlow, PyTorch і Keras.

Ключовою особливістю Python у контексті машинного навчання є його інтеграція з C-розширеннями через API CPython, що дає змогу оптимізувати виконання обчислювально інтенсивних операцій. Взаємодія Python з бібліотеками BLAS (Basic Linear Algebra Subprograms) та LAPACK (Linear Algebra Package) дозволяє ефективно вирішувати задачі оптимізації, які використовуються у стохастичних методах градієнтного спуску.

Завдяки доступу до низькорівневих бібліотек, таких як cuDNN та CUDA, Python забезпечує апаратне прискорення обчислень на GPU. Python підтримує обробку та аналіз великих обсягів даних через бібліотеки pandas, Dask та Vaex, які оптимізовані для паралельних обчислень і роботи з розрідженими структурами даних. У машинному навчанні Python виступає інтерфейсом до популярних фреймворків, таких як scikit-learn, що надає API для реалізації алгоритмів

класифікації, кластеризації та регресії, або XGBoost та LightGBM для побудови моделей градієнтного бустингу. Фреймворки TensorFlow і PyTorch використовують декларативний та імперативний підходи відповідно, забезпечуючи Python можливість створювати обчислювальні графи та динамічно оновлювати параметри моделей. Ці фреймворки також реалізують механізм автоматичного диференціювання (autograd), що важливо для навчання моделей із великою кількістю параметрів і складними функціями втрат. Модулі threading і multiprocessing дозволяють Python ефективно використовувати багатопоточність і багатопроцесорність, хоча блокування інтерпретатора може обмежувати продуктивність у багатопотоковому режимі.

4.1.2 Бібліотека Torch

Torch – це високопродуктивна обчислювальна бібліотека для роботи з тензорами, оптимізована для реалізації складних алгоритмів машинного навчання, яка інтегрується з Python і забезпечує ефективну підтримку роботи на CPU та GPU. Torch побудована на основі багатошарової архітектури, що включає низькорівневі обчислювальні операції, обгортки для лінійної алгебри, числових методів і нейронних мереж, що дозволяє використовувати її для широкого спектра ML-завдань, зокрема для навчання моделей на великих обсягах даних. Ключовою особливістю Torch є робота з багатовимірними тензорами (подібно до багатовимірних масивів у NumPy) та операції над ними, оптимізовані для паралельних обчислень із використанням сучасних GPU через CUDA або OpenCL. Torch реалізує автоматичну диференціацію через модуль torch.autograd, який динамічно обчислює градієнти будь-яких тензорних операцій, що критично важливо для навчання моделей глибинного навчання з використанням зворотного поширення помилки. Використання обчислювальних графів, що генеруються в реальному часі, забезпечує високу гнучкість у роботі з динамічними нейронними мережами. Torch включає модуль torch.nn, який надає інструменти для створення нейронних мереж за допомогою високорівневих абстракцій, таких як шари, функції

активації, регуляризація та нормалізація. Модуль torch.optim забезпечує широкий вибір оптимізаторів (*SGD, Adam, RMSprop*), які використовуються для пошуку оптимальних параметрів моделей. Torch також підтримує роботу зі збереженням та завантаженням моделей через torch.save і torch.load, що спрощує управління циклом моделей та інтеграцію в системи розгортання.

4.1.3 Бібліотека TorchVision

TorchVision – це бібліотека, побудована на основі PyTorch, яка забезпечує зручний інструментарій для роботи із задачами комп'ютерного зору. TorchVision інтегрує функціональність для попередньої обробки зображень, роботи з датасетами, готових моделей нейронних мереж та модулів для реалізації складних алгоритмів CV.

Бібліотека дозволяє спростити процес розробки систем глибинного навчання для аналізу зображень, використовуючи ефективну взаємодію з PyTorch для обчислення градієнтів і навчання моделей. Ключовий компонент TorchVision – це torchvision.transforms, модуль, який надає функції для перетворення зображень, такі як зміна розміру, нормалізація, відображення, повороти та додавання шуму. Ці трансформації забезпечують автоматизацію процесу доповнення даних (data augmentation), що покращує узагальнюючу здатність моделей. Для роботи з тензорами зображень TorchVision оптимізована під формат даних PyTorch, забезпечуючи високу продуктивність і ефективне управління пам'яттю.

TorchVision включає модуль torchvision.datasets, який надає доступ до популярних датасетів для задач комп'ютерного зору, таких як CIFAR, ImageNet, COCO, MNIST, та інші. Інтеграція з PyTorch DataLoader дозволяє ефективно розподіляти обчислювальні ресурси при підготовці великих обсягів даних. TorchVision також включає набір попередньо натренованих моделей у модулі torchvision.models, таких як ResNet, VGG, EfficientNet, MobileNet, DenseNet та трансформери, адаптовані до задач CV, наприклад Vision Transformer. Ці моделі дозволяють легко імпортувати архітектуру та ваги, попередньо навчені на великих

датасетах (наприклад, ImageNet), для їх використання у задачах transfer learning. TorchVision підтримує роботу з різними форматами зображень, зокрема PNG та JPEG, використовуючи модуль torchvision.io для завантаження та обробки мультимедійних даних [27].

4.1.4 Бібліотека Diffusers

Diffusers – це спеціалізована бібліотека, яка забезпечує інструментарій для роботи з дифузійними моделями. Бібліотека побудована на основі PyTorch і надає зручний API для реалізації, навчання та використання генеративних моделей, таких як Denoising Diffusion Probabilistic Models (DDPM), Latent Diffusion Models (LDM), та Stable Diffusion. Diffusers оптимізована для інтеграції з сучасними апаратними прискорювачами, такими як GPU, і забезпечує підтримку змішаної точності обчислень (mixed precision) для підвищення ефективності. Ключовий компонент Diffusers – це Pipeline, модуль, який надає готові конвеєри для виконання задач генерації. Конвеєри об’єднують усі необхідні компоненти: попередньо навчені моделі, текстові токенайзери (для мультимодальних задач), процедури дифузії та денойзингу.

Наприклад, модулі StableDiffusionPipeline, TextToImagePipeline чи ImageInpaintingPipeline дозволяють легко реалізувати генерацію зображень на основі текстових описів, редагування зображень чи їх відновлення. Бібліотека підтримує модульність компонентів, включаючи вибір оптимальних процедур дифузії, таких як DDIM або PNDM, які знижують кількість необхідних ітерацій для якісної генерації. Це забезпечує більш швидке виконання генеративного процесу, зберігаючи при цьому високу якість зображень. Diffusers інтегрується з фреймворками Hugging Face, такими як Transformers, що дозволяє використовувати текстові моделі (наприклад, CLIP чи GPT) для генерації мультимодальних даних. Завдяки цьому бібліотека підтримує складні сценарії генерації, наприклад, створення зображень, що відповідають заданому тексту, або стилізацію зображень відповідно до заданих описів. Модуль Scheduler реалізує алгоритми управління

процесом дифузії, які можна налаштовувати залежно від задачі. Наприклад, Linear Scheduler або Cosine Scheduler дозволяють адаптувати швидкість навчання до специфічних вимог.

4.1.5 Бібліотека Clip

CLIP – це бібліотека на Python, створена для роботи з моделями, що забезпечують спільний латентний простір для текстових та зображувальних представлень. Вона реалізована на базі PyTorch і забезпечує інструменти для завантаження, використання та тонкого налаштування попередньо навчених моделей, які поєднують текстові та візуальні енкодери. Текстовий енкодер зазвичай базується на трансформерах (наприклад, BERT), тоді як візуальний енкодер використовує сучасні архітектури для аналізу зображень, такі як ResNet або Vision Transformer (ViT). Бібліотека дозволяє працювати із задачами класифікації, пошуку та оцінки схожості між текстом і зображеннями, використовуючи підхід контрастивного навчання, який оптимізує представлення у спільному просторі шляхом максимізації косинусної подібності між правильними парами "текст-зображення" та мінімізації її для невідповідних пар. Однією з ключових особливостей CLIP є підтримка zero-shot класифікації, яка дозволяє класифікувати зображення без додаткового навчання моделі, використовуючи лише текстові дескриптори, наприклад, для визначення категорій зображень. Бібліотека також надає модулі для підготовки текстових і зображувальних даних, включаючи токенізацію тексту, нормалізацію зображень, зміну їх розмірів та перетворення до форматів, необхідних для моделей.

Інтеграція з PyTorch забезпечує ефективну підтримку обчислень на GPU, включаючи використання CUDA для пришвидшення навчання та інференсу.

4.2 Діаграма компонентів

Діаграма компонентів є одним із структурних видів діаграм у мові моделювання UML, яка використовується для візуалізації, специфікації, побудови та документування архітектури програмного забезпечення. Основна функція цієї діаграми полягає у моделюванні статичної структурної архітектури системи, з акцентом на компоненти, їхні інтерфейси, залежності між ними та механізми взаємодії. Компоненти на діаграмі є автономними модульними елементами системи, які представляють собою фізичні модулі коду, такі як бібліотеки, файли, пакети, веб-сервіси або інші логічні блоки.

На схемі, яка представлена в додатку Г, зображена діаграма компонентів системи генерації зображень з використанням ймовірісної моделі дифузії.

Система, відображена на діаграмі компонентів, складається із взаємопов’язаних модулів, кожен з яких виконує конкретну роль у процесі генерації зображень. Нижче наведено деталізований опис функціональності кожного компонента та їх призначення. Сховище навчальних даних виконує функцію централізованого сховища вхідних наборів даних, призначених для навчання нейронної мережі. У цьому модулі здійснюється збирання, зберігання та організація даних у форматі, оптимальному для подальшої обробки. Основними задачами компоненту є управління доступом до даних, забезпечення їхньої актуальності та підготовка для передачі у наступні модулі. Особлива увага приділяється ефективній роботі з великими обсягами даних за рахунок використання методів оптимізації зберігання.

Попередня обробка вхідних зображень відповідає за підготовку даних перед їх використанням у процесі навчання нейронної мережі. Включає операції зі стандартизації вхідних даних, такі як нормалізація розмірів, видалення шумів, корекція кольорів та перевірка якості зображень. Модуль автоматизує обробку великих обсягів даних, гарантуючи сумісність із вимогами наступних компонентів. Навчання моделі нейронної мережі реалізує процес створення та тренування моделі, яка буде використовуватись у процесі генерації зображень. Основні

завдання цього компонента включають ініціалізацію архітектури нейромережі, оптимізацію параметрів та ваг за допомогою алгоритмів навчання, а також збереження натренованих моделей для їх подальшого використання. Цей модуль забезпечує взаємодію з обробленими даними та підтримує гнучкість у виборі методик навчання.

Генерація зображень – ключовий компонент системи, який здійснює синтез зображень відповідно до заданих користувачем параметрів. Генерація відбувається на основі навченої моделі, яка отримує на вхід специфікації, що формують бажаний результат. Модуль підтримує гнучкість налаштувань, включаючи масштабування, деталізацію та вибір стилю генерації, а також забезпечує збереження отриманих результатів для подальшої обробки.

Постгенераційне покращення зображень відповідає за підвищення якості згенерованих зображень. Використовуючи попередньо навчену модель (наприклад, декодер), модуль виконує деталізацію, покращення текстур, корекцію кольорової гами та підвищення роздільної здатності. Додатково здійснюється перевірка на відповідність вихідних зображень заданим стандартам якості. Перетворення зображень виконує функції адаптації вихідних даних до різних платформ та форматів. Цей модуль забезпечує масштабування, перетворення форматів, додавання метаданих та оптимізацію зображень для відображення чи подальшого використання.

Окремою задачею є уніфікація вихідних даних відповідно до специфікацій різних середовищ. Інтерактивний інтерфейс користувача забезпечує зв’язок між користувачем та системою. Цей модуль відповідає за збирання необхідних параметрів для генерації зображень, таких як текстовий опис, характеристики візуалізації, розмір зображення тощо. Крім того, він відображає прогрес виконання задачі, результати генерації та дозволяє змінювати параметри в режимі реального часу. Компонент також виконує функції інтеграції всіх модулів системи, забезпечуючи їх узгоджену роботу. Система базується на чіткій взаємодії між компонентами, кожен із яких передає вихідні дані у форматі, необхідному для роботи наступного модуля.

4.3 Діаграма варіантів використання

Діаграма варіантів використання (прецедентів) – це графічне представлення взаємодії акторів із системою через її функціональність, де кожен прецедент відображає конкретну ціль або задачу, яку виконує система для досягнення певного результату, із позначенням зв'язків між актором і прецедентами, що забезпечує розуміння функціональних вимог, меж системи та ролей користувачів без деталізації реалізації чи технічних аспектів, слугуючи основою для специфікації вимог і подальшого проектування.

Діаграма варіантів використання системи представлена в додатку В.

У системі визначено два актори: анонімний користувач та авторизований користувач. Анонімний користувач має можливість зареєструватися у системі, для чого передбачено введення особистих даних, після чого виконується авторизація, яка включає в себе вхід до системи.. Авторизований користувач може вводити текстові запити, які слугують основою для генерації зображень, налаштовувати параметри генерації, такі як розмір чи стиль, переглядати результати роботи системи у вигляді згенерованих зображень і зберігати ці результати для подальшого використання. Діаграма також відображає логічні залежності між варіантами використання, наприклад, авторизація користувача включає в себе етап входу до системи, а введення текстового запиту є обов’язковим для генерації.

4.4 Діаграма послідовності

Діаграми послідовності в UML є одним із видів поведінкових діаграм, які використовуються для моделювання взаємодії між об’єктами системи в рамках певного сценарію. Вони фокусуються на часовій послідовності повідомлень, які передаються між об'єктами, і демонструють, як виконується певний процес або функція. Основна мета таких діаграм – деталізувати порядок виконання дій і показати, як компоненти системи взаємодіють один з одним. Діаграма послідовності складається з кількох ключових елементів. Учасники та об’єкти

відображаються у вигляді вертикальних ліній, які називаються лініями життя. Ці лінії представляють різні сутності системи, наприклад, користувачів, інтерфейси, сервери або бази даних. Горизонтальні стрілки між лініями життя позначають повідомлення, які можуть бути викликами методів, запитами чи відповідями. Повідомлення розташовуються відповідно до порядку виконання, що дозволяє візуалізувати хронологію.

У додатку Е представлена діаграма послідовності процесу реєстрації, яка демонструє взаємодію між різними об'єктами: користувачем, інтерфейсом, сервером та локальною базою даних. Послідовність взаємодій починається з того, що користувач вводить дані реєстрації через інтерфейс. інтерфейс передає ці дані на сервер для обробки. Сервер, у свою чергу, виконує перевірку наявності користувача в локальній базі даних шляхом надсилання запиту. Якщо користувач не знайдений, сервер отримує відповідь від бази даних про відсутність збігів. Після цього сервер зберігає нові дані користувача в локальній базі даних, отримуючи підтвердження успішного збереження.

Після завершення цього етапу сервер надсилає повідомлення інтерфейсу про успішне завершення реєстрації, яке відображається користувачу.

4.5 Діаграма діяльності

Діаграми діяльності в UML є інструментом для моделювання поведінки системи, що дозволяє описувати послідовність виконання дій, потоки управління та передачі даних у процесах. Ці діаграми представляють собою орієнтовані графи, де вузли позначають дії, рішення, початок або завершення процесу, а дуги – потоки управління чи передачу об’єктів.

Основні компоненти діаграми включають дії, які виконуються системою, контрольні потоки, що визначають порядок виконання дій, розгалуження і злиття для реалізації умовної логіки, а також вузли паралелізму, які моделюють одночасне виконання декількох потоків. Крім цього, діаграми підтримують об’єктні потоки для показу передачі даних між різними діями. Цей інструмент широко

використовується для візуалізації та документування складних алгоритмів, бізнес-процесів і робочих потоків.

У додатку З представлена діаграма діяльності системи генерації зображень, яка відображає покроковий процес роботи системи, починаючи від підготовки навчальних даних до отримання вихідного результату у вигляді зображень. Процес починається з набору навчальних зображень, які завантажуються в систему. Після завантаження виконується первинна обробка зображень, у результаті чого отримуються оброблені дані, які готуються для використання у подальших етапах.

Далі оброблені зображення використовуються для ініціалізації моделі, яка проходить навчання. У цьому процесі створюється навчена дифузійна модель, що є основою для генерації зображень. Паралельно виконується екстракція моделі, що передається на етап генерації. У процесі генерації створюються згенеровані дані, які можуть бути додатково покращені за допомогою моделі покращення зображень, результатом чого є покращені зображення. Наступним етапом є конвертація даних у зображення.

Після цього система забезпечує попередній перегляд результатів для оцінки якості та точності роботи. Завершальний етап полягає в отриманні вихідного зображення, яке передається як кінцевий результат роботи системи.

4.4 Розробка компонентів системи

4.4.1 Завантаження та попередня обробка даних

Ця частина реалізації призначена для підготовки даних, необхідних для навчання генеративних дифузійної моделі. Її основна мета – завантаження, попередня обробка та подання зображень у форматі, придатному для навчання, з можливістю обробки класів для умовного моделювання. Код реалізує функцію load_data, яка є головним інтерфейсом для створення генератора пар даних (зображень та відповідних параметрів), а також допоміжні функції та класи для роботи з набором даних. Функція load_data відповідає за рекурсивне сканування вказаного каталогу (data_dir) для збору всіх файлів зображень у підтримуваних

форматах (JPEG, PNG, GIF). Для цього використовується допоміжна функція _list_image_files_recursively, яка перевіряє файли у каталозі, додаючи їхні шляхи до списку, та рекурсивно обробляє вкладені папки. Далі, на основі зібраного списку зображень, створюється об'єкт ImageDataset. Цей клас є нащадком torch.utils.data.Dataset і призначений для завантаження та попередньої обробки зображень. Він підтримує поділ даних на фрагменти (шарди) для розподіленої обробки за допомогою MPI. Кожен процес MPI отримує лише свою частину даних (визначену параметрами shard та num_shards), що забезпечує паралельне завантаження. У класі ImageDataset реалізований метод __getitem__, який завантажує зображення з файлу, виконує попередню обробку, включаючи зміну розміру до заданого роздільного здаття (resolution), і перетворює його у нормалізований формат тензора (зі значеннями в діапазоні [-1, 1]). Зміна розміру зображення виконується за допомогою методів бібліотеки PIL, включаючи послідовне зменшення розміру (якщо зображення занадто велике) та масштабування зі збереженням пропорцій. Після масштабування виконується центральне обрізання до кінцевого розміру.

Оброблене зображення перетворюється у формат NCHW (канали, висота, ширина), який використовується бібліотекою PyTorch.

Якщо класи були визначені, вони додаються до словника (out_dict), який повертається разом із зображенням.

4.4.2 Імплементація процесів дифузії

Клас GaussianDiffusion є комплексною абстракцією, яка реалізує математичну модель дифузійного процесу для генерації даних. Його архітектура дозволяє інтегрувати різні типи β-графіків, обчислювати ключові статистичні параметри процесу, а також виконувати функції генерації та обчислення втрат. До атрибутів класу GaussianDiffusion можна віднести:

– betas: Одновимірний масив β-коефіцієнтів для кожного часового кроку дифузії. Ці параметри визначають швидкість додавання шуму в процесі прямої дифузії.

– alphas_cumprod: Кумулятивний добуток 1 - β, який визначає ступінь збереження інформації про початковий сигнал на кожному кроці.

– posterior_variance: Дисперсія апостеріорного розподілу, яка моделює ймовірність переходу від одного кроку дифузії до попереднього.

– model_mean_type, model_var_type, loss_type: Перелік вибору типів середніх значень, дисперсій та втрат, які визначають, як модель обробляє дані та передбачає результати.

Метод q_mean_variance обчислює характеристики розподілу $q(x_t|x_0)$ який представляє шумові дані після t t-го кроку. Він приймає як аргументи початкові дані x_0 та часовий індекс t. На основі попередньо обчислених параметрів (таких як sqrt_alphas_cumprod і log_one_minus_alphas_cumprod) метод розраховує середнє, дисперсію та логарифм дисперсії для розподілу.

Метод q_sample реалізує прямий дифузійний процес $q(x_t|x_0)$. Він додає шум до початкового сигналу x_0, використовуючи заданий часовий індекс t і, за необхідності, зовнішньо заданий шумовий сигнал. У разі, якщо шум не заданий явно, він генерується як випадковий гаусівський шум. Метод використовує параметри $\sqrt{a}$ і $\sqrt{1-a}$ для точного додавання шуму до даних.

Метод q_posterior_mean_variance обчислює характеристики апостеріорного розподілу $q(x_{t-1}|x_t, x_0)$. Цей розподіл використовується для моделювання зворотного процесу, що відновлює початкові дані з шуму. Метод обчислює середнє значення та дисперсію апостеріорного розподілу, використовуючи параметри β, a та їхні кумулятивні добутки.

Метод p_mean_variance відповідає за моделювання зворотного процесу $p(x_{t-1}|x_t)$ який передбачає, як шум на поточному кроці t t перетворюється у попередній крок $t-1$. Він також виконує передбачення початкового сигналу через модель. У разі, якщо передбачення моделі є некоректним, реалізовано механізм

кліпінгу (обмеження значень) до діапазону [-1, 1]. Метод підтримує різні типи виводу моделі: попередній сигнал x_{t-1} та початковий сигнал x_0.

Метод p_sample реалізує стохастичний семплінг зворотного процесу $p(x_{t-1}|x_t)$. Використовуючи середнє значення та дисперсію, обчислені в p_mean_variance, метод додає випадковий шум, щоб забезпечити розмаїття зразків. Важливо, що цей метод враховує часовий індекс t, а також дозволяє використовувати функції для додаткового оброблення даних, якщо потрібно.

Метод p_sample_loop виконує повний семплінг даних, починаючи з шумового стану x_T і поступово "відмотуючи" процес назад до початкового стану x_0. Він ітерує через всі часові індекси, викликаючи метод p_sample на кожному кроці. У результаті створюється семпл, який імітує зображення.

Метод ddim_sample є детермінованим варіантом семплінгу, який використовує підхід DDIM. Цей метод дозволяє контролювати рівень детермінованості процесу через параметр η. На відміну від класичного стохастичного семплінгу, DDIM забезпечує більш стабільні результати, особливо для задач, що вимагають точного відновлення.

Метод training_losses виконує обчислення втрат для кожного кроку навчання. Він підтримує різні типи функцій втрат, такі як середньоквадратична похибка (MSE) та KL-дивергенція. У випадку, якщо дисперсія є навчуваною, метод враховує цей аспект, обчислюючи відповідні терміни втрат. Крім того, він може інтегрувати додаткові параметри, такі як кліпінг значень або врахування апостеріорного розподілу.

Метод _vb_terms_bpd розраховує компоненти варіаційної нижньої межі (VLB) для кожного кроку. Він обчислює KL-дивергенцію між справжнім і передбаченим розподілом, а також оцінює ентропію для задач семплінгу.

Метод calc_bpd_loop виконує обчислення всієї варіаційної нижньої межі для всього дифузійного процесу. Він повертає як загальну величину, так і детальні компоненти, включаючи втрати на кожному часовому кроці. Це є інструментом для оцінки якості моделі та її генеративних можливостей.

4.4.3 Тренування моделі

Схема архітектури нейронної мережі U-Net у моделі ймовірісної дифузії представлена у додатку Ж. Розроблена модель має ієрархічну структуру, що складається з блоків для послідовного зменшення роздільної здатності (downsampling), обробки в найнижчій точці (bottleneck), а потім збільшення роздільної здатності (upsampling). Ключовими компонентами є залишкові блоки , блоки уваги, а також механізми обробки тимчасових кроків.

1.1. Вхідні дані x проходять через перший вхідний блок (input_blocks), який виконує початкову обробку, використовуючи 3x3 згортки.

1.2. Якщо модель є клас-контекстуальною, додається вектор тимчасової ембеддингу (time embedding), що інформує мережу про поточний крок дифузії.

2. Зменшення роздільної здатності (Downsampling):

2.1 Вхідні дані проходять через кілька рівнів input_blocks, кожен з яких містить залишкові блоки (ResBlock) і, за потреби, блоки уваги (AttentionBlock).

2.2 Кожен рівень зменшує просторову роздільну здатність даних через шар Downsample, що реалізується за допомогою середнього пулінгу або згортки з кроком 2.

3. Середній блок (Bottleneck):

3.1 У найнижчій точці обробки (middle_block) дані проходять через два залишкові блоки (ResBlock) і шар уваги (AttentionBlock).

3.2 Цей шар дозволяє моделі обробляти глибокі представлення даних із найбільш зменшеною роздільною здатністю.

4. Збільшення роздільної здатності (Upsampling):

4.1 Дані проходять через output_blocks, які включають залишкові блоки, блоки уваги та механізми апсемплінгу (Upsample).

4.2 На кожному рівні дані з попередніх етапів (через конкатенацію з прихованими станами з input_blocks) допомагають зберегти високорівневу інформацію.

5. Вихідний шар:

5.1 Після обробки даних у всіх рівнях, дані передаються через нормалізацію, (SiLU) та останню 3x3 згортку для формування остаточного результату.

Метод ініціалізації __init__ створює всю архітектуру моделі. Основні параметри включають:

– in_channels, out_channels – кількість каналів у вхідному та вихідному тензорах.

– model_channels – базова кількість каналів у моделі.

– num_res_blocks – кількість залишкових блоків на кожному рівні.

– dropout, channel_mult, conv_resample, dims, num_classes, use_checkpoint, num_heads – додаткові параметри для налаштування мережі, таких як використання дропаута, кількість голів уваги та градієнтне контрольоване обчислення (checkpointing).

Для тренування моделі було прийняте рішення використовувати Google Colab прийняте через необхідність значних обчислювальних ресурсів для навчання моделі. Навчання дифузійних моделей є вкрай ресурсозатратним процесом, що потребує доступу до GPU чи TPU, які пропонує Google Colab, забезпечуючи баланс між продуктивністю і доступністю, зокрема для проведення експериментів чи виконання довготривалих обчислень. В першу чергу ініціалізується об'єкт Accelerator, який дозволяє оптимізувати процеси обчислення, враховуючи змішану точність (mixed precision), накопичення градієнтів (gradient accumulation) і ведення логів за допомогою TensorBoard. Директорії для збереження логів та вихідних даних створюються залежно від конфігурації, а при необхідності може бути створене репозиторій у Hugging Face Hub для автоматичної публікації моделі. На етапі підготовки об'єкти, включаючи модель, оптимізатор, завантажувач даних (train_dataloader) та планувальник швидкості навчання (lr_scheduler), передаються методу accelerator.prepare, що налаштовує їх для ефективної роботи на доступних апаратних засобах (CPU або GPU). Цикл навчання розбивається на епохи, кількість яких задається конфігурацією. У кожній епосі обробляється весь набір навчальних даних, причому прогрес відображається через tqdm, якщо процес є локальним

головним. На кожному кроці батч даних містить "чисті" зображення, до яких додається випадковий шум, симулюючи процес дифузії.

Випадково обрані часові кроки визначають рівень шуму, що додається до кожного зображення, а потім використовується модель для передбачення залишкового шуму. Втрати обчислюються як середньоквадратична помилка між передбаченим шумом та реальним, після чого вони передаються для зворотного поширення градієнтів через accelerator.backward. Логи, що включають втрати, значення швидкості навчання та глобальний крок, передаються в TensorBoard через accelerator.log. Після завершення епохи основний процес може створити демонстраційні зображення, використовуючи оцінювальну функцію, і за потреби зберегти модель.

Збереження відбувається як локально, так і через Hugging Face Hub залежно від налаштувань конфігурації.

На рисунку 4.3 представлено процес тренування моделі мережі.

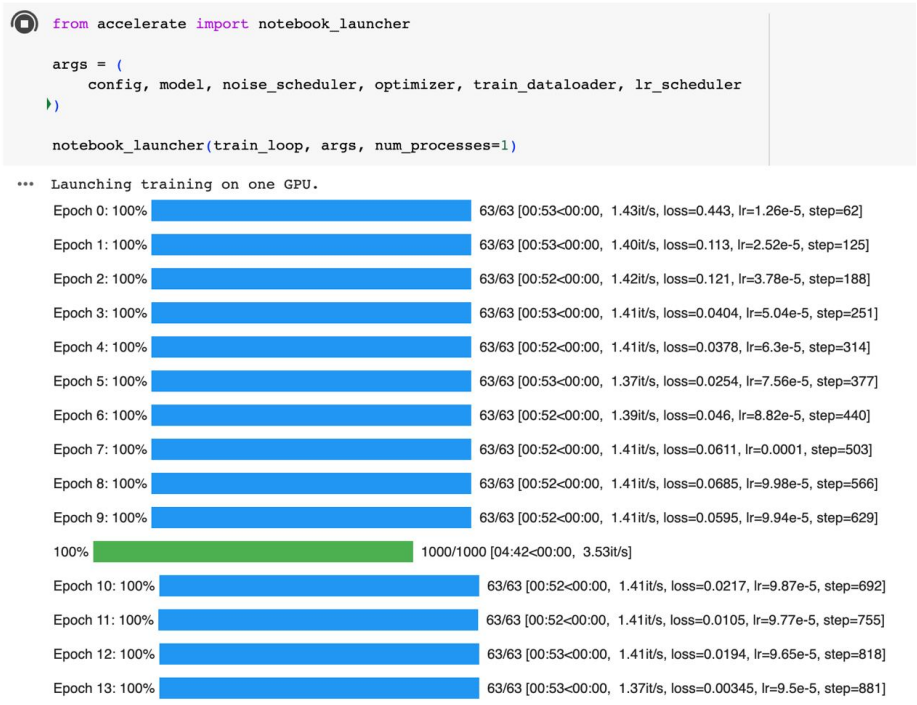

Рисунок 4.3 – Процес тренування моделі нейронної мережі

На рисунку 4.4 представлена динаміка зміни двох основних параметрів під час тренування моделі: втрат (loss) та швидкості навчання (learning rate, lr).

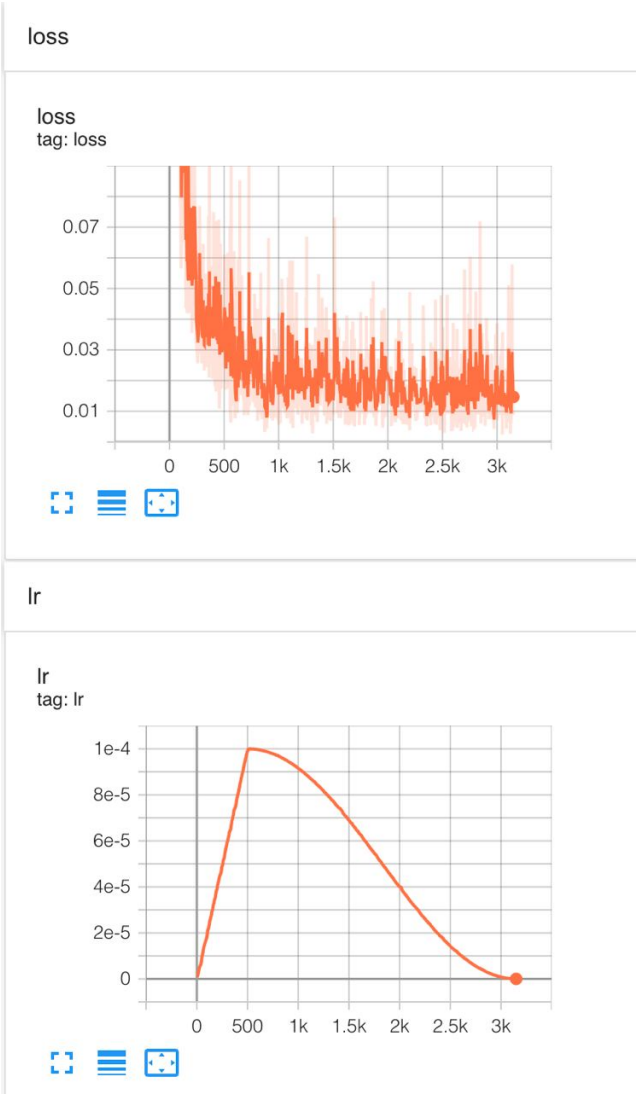

Рисунок 4.4 – Динаміка зміни параметрів втрат (loss) та швидкості навчання (learning rate, lr).

На першому графіку, позначеному як "loss", відображено зниження функції втрат у процесі навчання. По осі абсцис (x) відкладається кількість ітерацій, а по осі ординат (y) – значення втрат. З початку навчання функція втрат має високі значення, що свідчить про початково невідповідну якість передбачень моделі. У процесі тренування спостерігається поступове зниження втрат, що демонструє ефективність навчання та здатність моделі адаптуватися до задачі. Однак після перших 500–1000 ітерацій можна помітити, що значення втрат коливаються навколо певного рівня, не показуючи суттєвого зменшення. Це може свідчити про досягнення стабільності моделі або про те, що навчання наближається до зони оптимуму. Другий графік, позначений як "lr" (learning rate), відображає зміну

швидкості навчання оптимізатора протягом тренувального процесу. По осі абсцис (x) також відкладається кількість ітерацій, а по осі ординат (y) – значення швидкості навчання. Графік демонструє, що швидкість навчання поступово збільшується на початкових етапах тренування, досягаючи пікового значення приблизно після 500–800 ітерацій. Після цього відбувається поступове зниження швидкості навчання, що відповідає стандартним стратегіям адаптивного зменшення learning rate (циклічний графік або cosine decay).

На рисунку 4.5 представлено результат генерації зображення з використанням натренованої моделі з поетапним знешумленням.

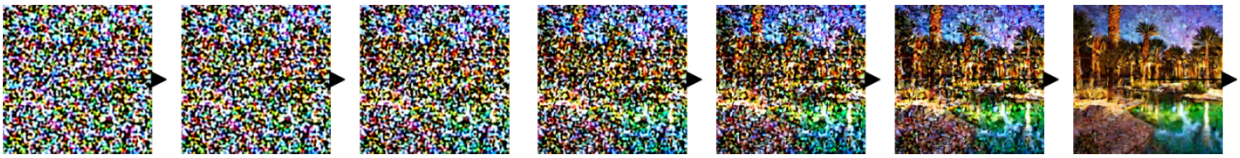

Рисунок 4.5 – Результат генерації зображення з поетапним знешумленням

Як видно з рисунку 4.5, цей процес демонструє перехід від повністю зашумленого зображення до високоякісного реконструйованого зображення за допомогою багатоетапного зменшення шуму. На першому етапі спостерігаємо зображення, яке повністю складається з випадкового шуму. Це зашумлене зображення є результатом дифузійного процесу, під час якого до оригінального зображення додавали шум на кожному часовому кроці, поступово розмиваючи його інформацію. Таким чином, перше зображення представляє останній крок цього процесу, де вся початкова структура втрачається, і зображення виглядає як набір випадкових кольорових пікселів. Далі показані послідовні кроки зворотного процесу, який називається процесом денойзингу. На кожному з цих кроків модель передбачає та зменшує рівень шуму, використовуючи свої навчені параметри. Вхідними даними для кожного кроку є зашумлене зображення та інформація про поточний часовий індекс. Алгоритм денойзингу включає прогнозування шумового компонента, який додається до зображення на цьому етапі, і його корекцію для відновлення початкових ознак. На кожному наступному зображенні на ілюстрації

ми бачимо, як поступово проявляються елементи структури та кольорів, які наближають зображення до його фінального вигляду. Спочатку починають з'являтися загальні контури, потім додаються більш чіткі деталі, такі як форми об'єктів, текстури та кольори. На кожному кроці рівень шуму зменшується, а модель реконструює втрачену інформацію з високою точністю. На останньому етапі процес завершується, і отримуємо повністю відновлене зображення.

Висновки до розділу 4

У четвертому розділі детально розглянуто процеси розроблення програмного забезпечення для системи генерації зображень на основі ймовірісних моделей дифузії, починаючи від завантаження та попередньої обробки даних до тренування моделі та оцінки її результатів. Було реалізовано повноцінний програмний цикл, який включає підготовку вхідних даних, створення механізмів дифузійних перетворень, архітектуру нейронної мережі та алгоритми навчання. Попередня обробка даних забезпечила ефективну підготовку великомасштабних наборів зображень для навчання, включаючи рекурсивне завантаження, нормалізацію та зміну роздільної здатності. Зокрема, розроблені методи дозволяють виконувати обчислення апостеріорних розподілів, передбачення початкових сигналів і зменшення шуму за допомогою як стохастичних, так і детермінованих підходів . Особлива увага приділена обчисленню втрат, що включає оцінку варіаційної нижньої межі для оптимізації моделі. Для навчання було використано платформу Google Colab, яка забезпечила необхідну продуктивність GPU для виконання ресурсоємних операцій. У процесі навчання застосовувались методики оптимізації швидкості навчання та ведення логів для моніторингу втрат і прогресу навчання. Ефективність навчання підтверджується графіками зменшення втрат, які демонструють поступову стабілізацію моделі, а також результатами генерації зображень, де показано поетапне відновлення якісних зображень із зашумлених даних. Використання дифузійної архітектури забезпечило високу здатність до генерації реалістичних зображень.

5 ТЕСТУВАННЯ СИСТЕМИ ГЕНЕРАЦІЇ ЗОБРАЖЕННЯ З ВИКОРИСТАННЯМ МОДЕЛІ ДИФУЗІЇ

5.1 Огляд функцій системи

Під час запуску системи користувачу буде надано можливість пройти автентифікацію (за умови, що він вже зареєстрований у системі) або виконати процедуру реєстрації. Ці етапи продемонстровано на рисунках 5.1 – 5.2.

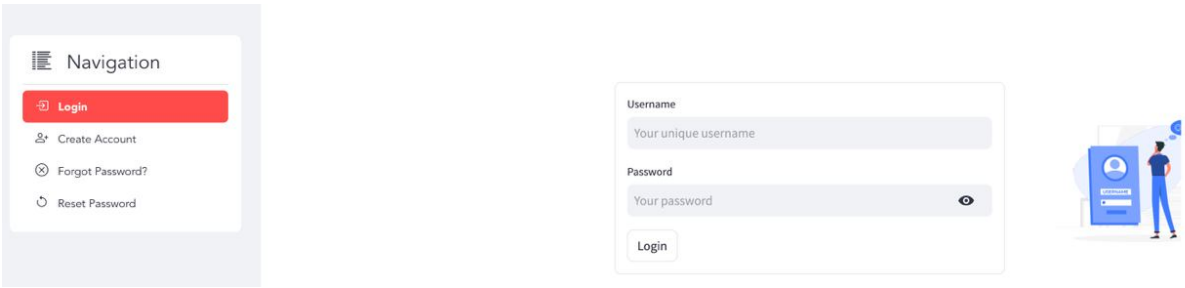

Рисунок 5.1 – Сторінка для входу в систему

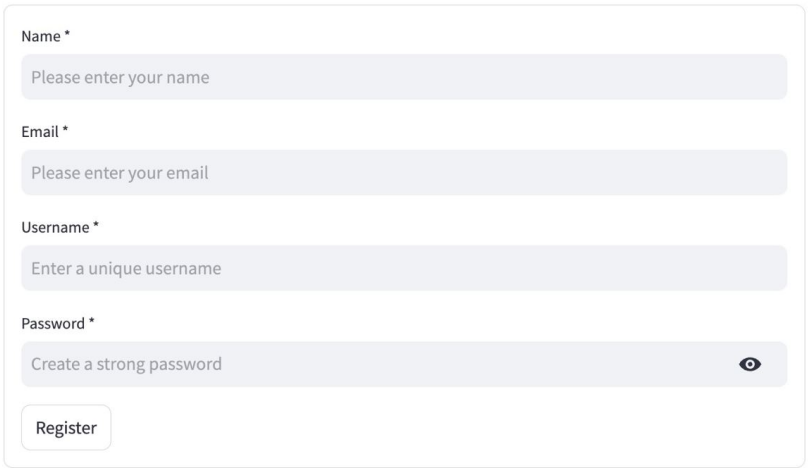

Рисунок 5.2 – Форма реєстрації в системі

Рисунок 5.3 демонструє сторінку, яка відкривається користувачу після успішного проходження автентифікації в системі.

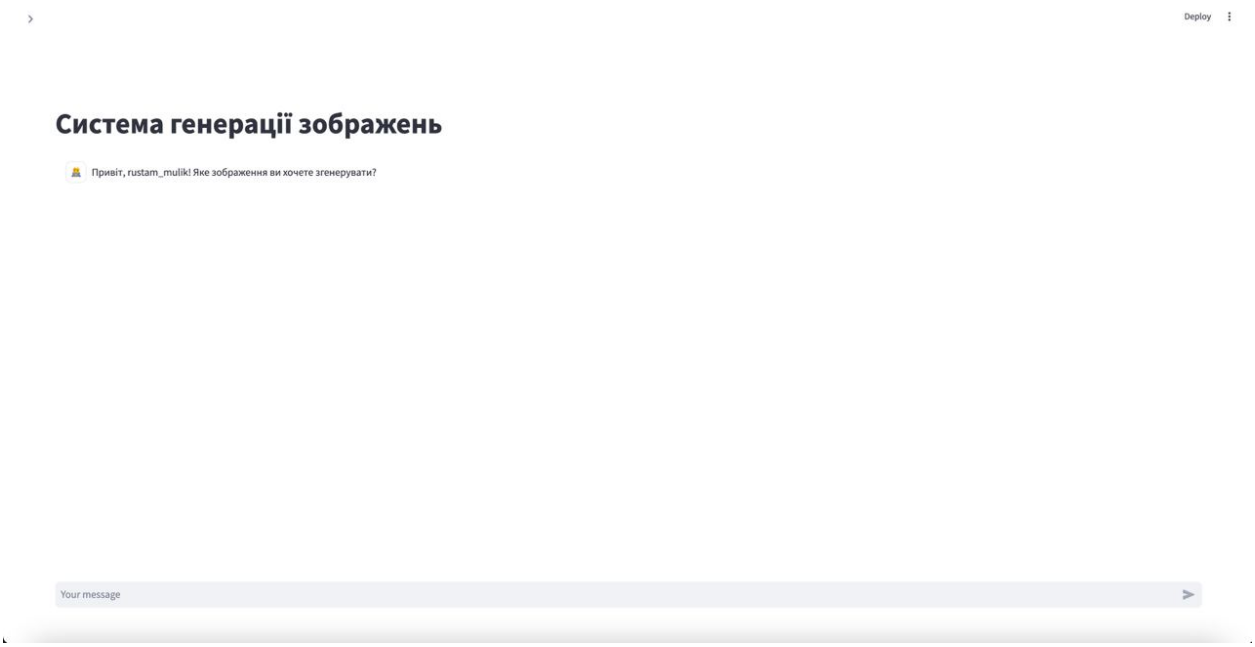

Рисунок 5.3 – Головна сторінка системи

На рисунку 5.4 продемонстровано процес генерації зображення, під час якого система динамічно відображає прогрес виконання. Цей етап розпочинається після того, як користувач вводить текстовий запит. Система аналізує введені дані та поступово формує результат, надаючи користувачу зручний інтерфейс для спостереження за ходом генерації в реальному часі.

Рисунок 5.4 – Процес генерації зображення за запитом користувача

На рисунку 5.5 представлно результат згенерованого зображення за наступним запитом користувача: “Кіт у капелюсі чарівника помішує казан у затишному котеджі”.

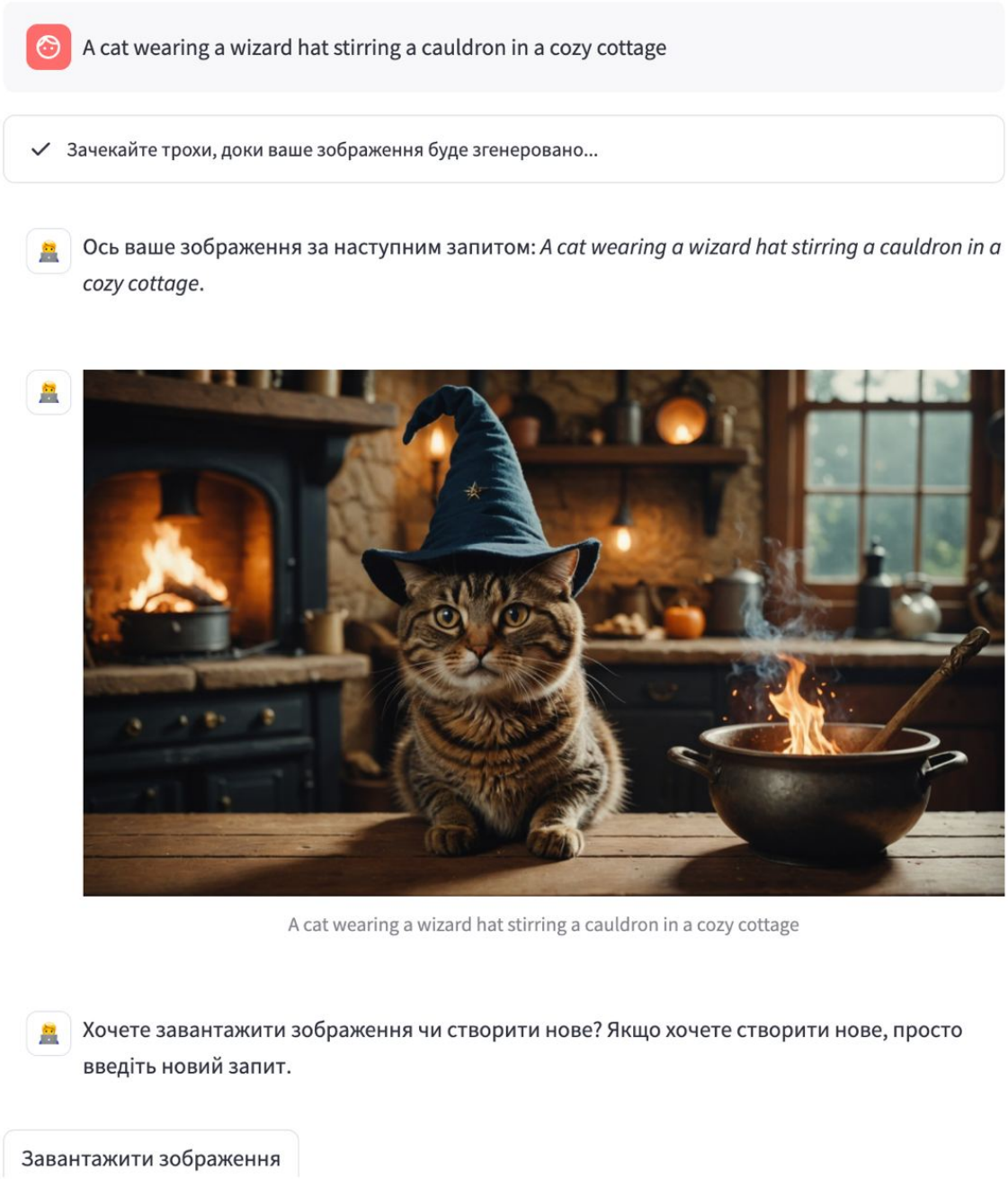

Рисунок 5.5 – Результат генерації зображення за текстовим користувача

Отже, система успішно згенерувала візуалізацію, яка відповідає заданому опису. На предсставлено зображенні система інтерпретувала запит "A cat wearing a wizard hat stirring a cauldron in a cozy cottage" і створила високодеталізовану сцену.

На зображенні представлений кіт у чарівницькому капелюсі, який взаємодіє з об’єктом, виконуючи контекстну дію – перемішування. Отриманий результат демонструє здатність системи точно інтерпретувати текстовий запит і створювати зображення, яке відповідає очікуванням користувача. Також, зображення можна завантажити.

На рисунках 5.6 – 5.7 представлені додаткові приклади згенерованих зображень за текстовим запитом користувача.

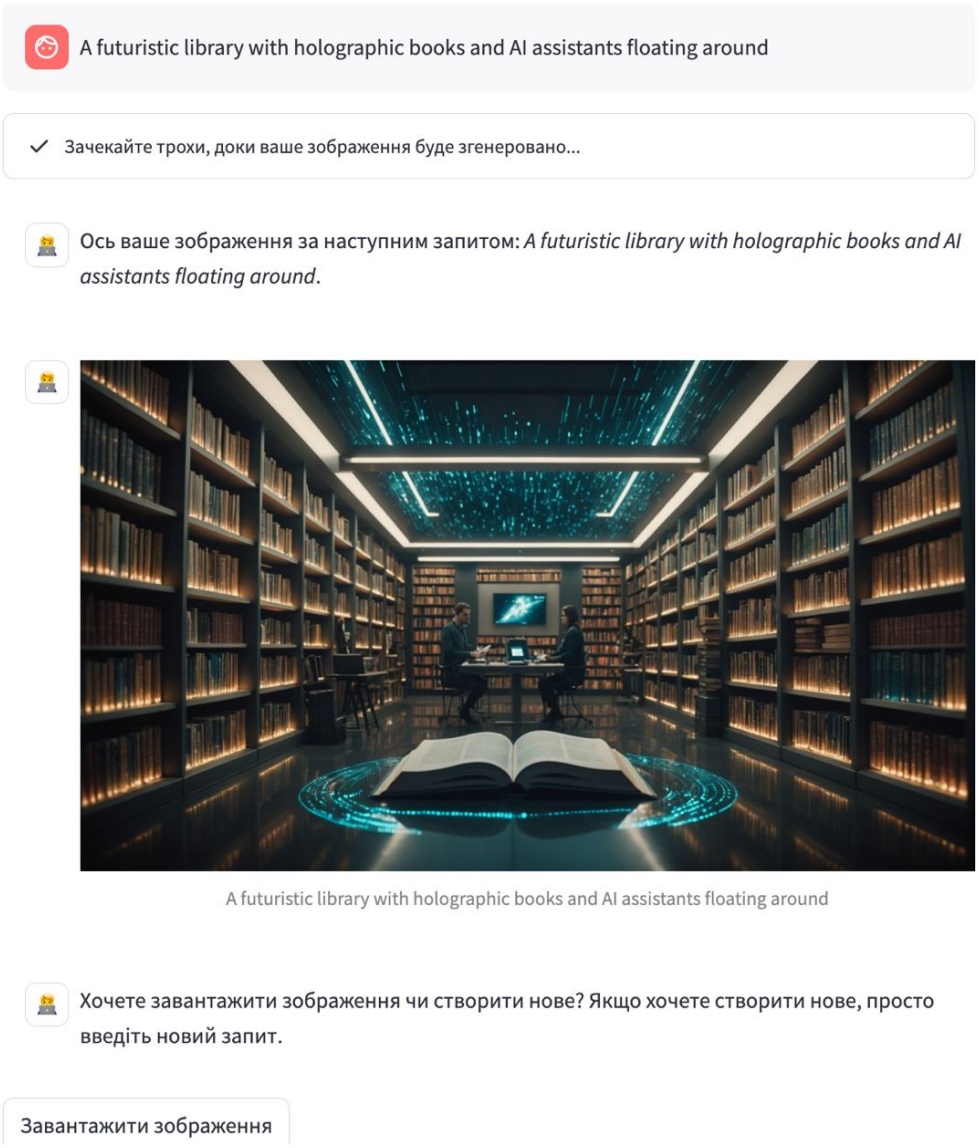

Рисунок 5.6 – Додатковий приклад результату згенерованого зображення за запитом користувача

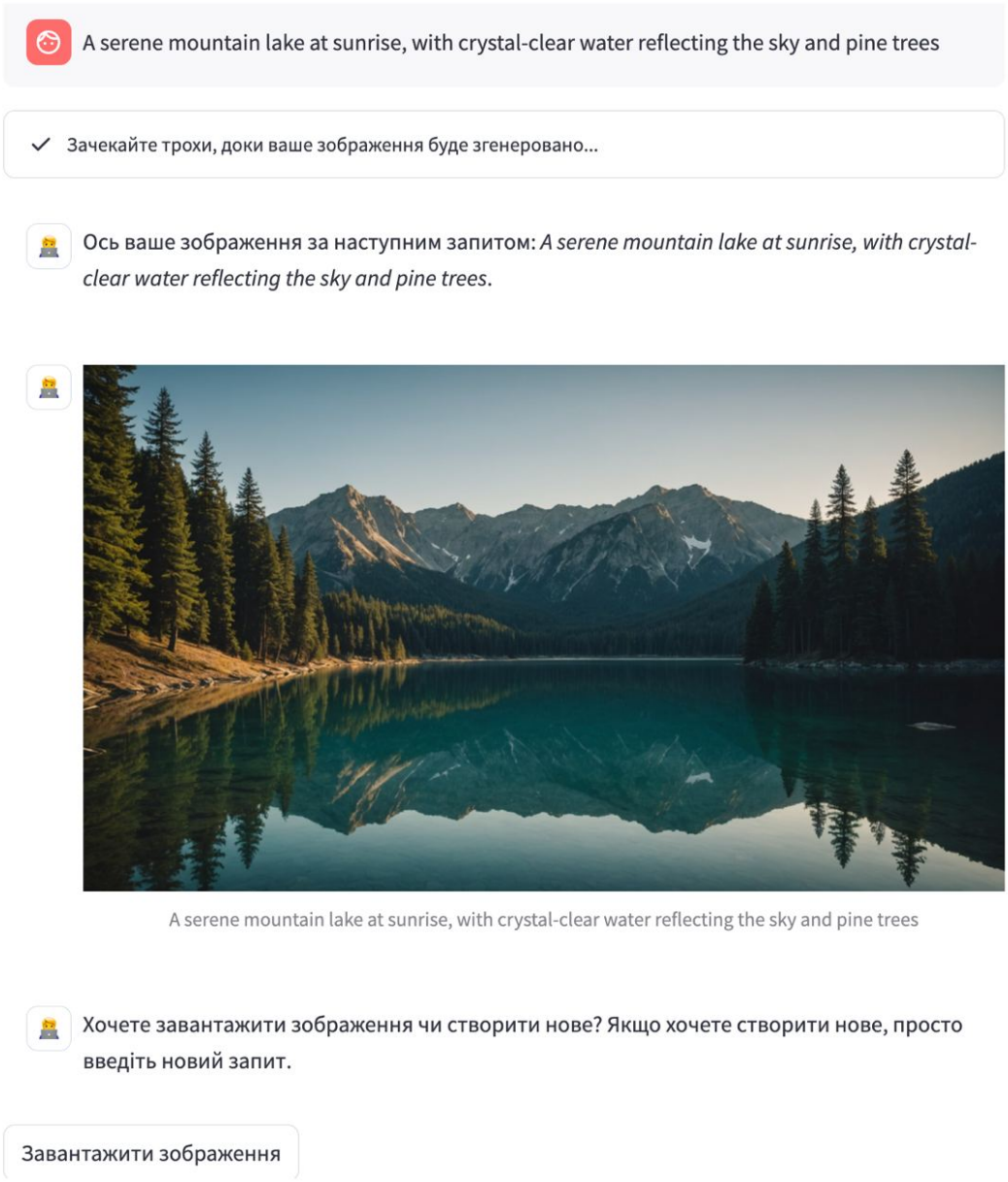

Рисунок 5.7 – Додатковий приклад результату згенерованого зображення за запитом користувача

На першому прикладі (рис. 5.6) система згенерувала футуристичну бібліотеку з голографічними книгами та помічниками. Зображення підкреслює атмосферу з сучасним дизайном бібліотеки та деталізованими елементами голограм. У другому прикладі (рис 5.7) система відобразила гірське озеро на світанку, де чиста вода відображає небо і соснові дерева.

Обидва приклади підтверджують здатність системи точно відповідати на текстові описи, створюючи зображення високої візуальної якості.

Висновки до розділу 5

У п’ятому було проведено тестування системи генерації зображень, яке підтвердило її функціональність, зручність використання та здатність точно інтерпретувати текстові запити. Система забезпечує автентифікацію та реєстрацію користувачів, пропонує інтуїтивний інтерфейс для взаємодії та відображає прогрес генерації в реальному часі. Представлені результати демонструють високу якість створених зображень, відповідність текстовим описам і деталізацію візуальних елементів.

Середній час генерації 11.5 секунд свідчить про ефективність алгоритму, а додаткові приклади підтверджують універсальність системи для різних текстових запитів.

6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

Головною метою цього розділу є ідентифікація найбільш ефективних шляхів інтеграції продукту на ринок, а також окреслення необхідних заходів, спрямованих на забезпечення його успішної комерціалізації та позиціонування в конкурентному середовищі.

6.1 Опис ідеї проєкту

В таблиці 6.1 описано ідеї проєкту, вигоди для користувачів даного рішення, а також напрямки застосування цього проєкту.

Таблиця 6.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка інноваційної системи, яка дозволяє генерувати високоякісні зображення на основі текстових описів, використовуючи алгоритми ймовірнісної дифузії.	Система може використовуватися для автоматизованого створення рекламних матеріалів, банерів, ілюстрацій для маркетингових кампаній, що значно спрощує роботу дизайнерів та зменшує витрати на їхні послуги.	Користувачі отримують можливість створювати високоякісний контент без необхідності залучення професійних дизайнерів, що значно скорочує витрати.
	Інструмент стане корисним для створення наочних матеріалів, таких як схеми, візуалізації складних процесів або навчальних концепцій, що дозволить покращити якість навчального контенту.	Інструмент забезпечує швидку генерацію візуалізацій, економлячи час для компаній та індивідуальних користувачів.
	Система допоможе у створенні	Навіть без спеціальних навичок користувачі

	персоналізованих візуалізацій для товарів, які ще не виготовлені, або покращити презентацію існуючих продуктів, підвищуючи їхню привабливість для споживачів.	можуть створювати кастомізовані зображення, що відповідають їхнім потребам і задумам. Завдяки простому та зрозумілому інтерфейсу, система дозволяє мінімізувати людський фактор у створенні зображень.

Після формулювання фундаментальної концепції системи, визначення сфер її потенційного застосування та ключових вигод для кінцевих користувачів, необхідно провести ретельний аналіз техніко-економічних параметрів запропонованої ідеї у контексті її порівняння з існуючими ринковими аналогами та конкурентними рішеннями. У наступній таблиці представлено порівняльну оцінку характеристик розроблюваної системи, де символ W (weak) позначає її слабкі сторони, N (neutral) – нейтральні аспекти, а S (strong) – сильні позиції. Декомпозиція параметрів у розрізі сильних, нейтральних і слабких сторін наведена у таблиці 6.2, що дозволяє комплексно оцінити конкурентоспроможність ідеї у заданих ринкових умовах.

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№	Технічні та економічні характеристики	Потенційні конкуренти та їх можливі продукти				W	N	S
1		Власна розробка (FastDPM)	DALL-E 2	MidJourney	Runway			
1	Має безкоштовний функціонал	+	+	-	-		+	
2	Використовує сучасні	+	+	+	-		+	

	дифузійні моделі							
3	Має зручний та інтуїтивний інтерфейс	+	+	-	+	+		
4	Забезпечує високу точність генерації зображень	+	+	+	-	+		
5	Дозволяє кастомізувати параметри генерації	-	+	+	-			+
6	Потребує невеликих обчислювальних ресурсів	+	+	+	+		+	
7	Забезпечує стабільну роботу при великих навантаженнях	+	+	+	+		+	

Результати порівняльного аналізу свідчать про те, що розроблюваний проєкт демонструє конкурентоспроможність за більшістю ключових характеристик, а за окремими параметрами перевершує існуючі аналоги.

Це створює вагомі передумови для успішного виходу проєкту на ринок.

6.2 Технологічний аудит ідей проєкту

Оцінка технічної здійсненності концепції може бути реалізована через проведення її детального технічного аудиту. Цей процес дозволяє визначити, наскільки реалізація ідеї є технічно обґрунтованою та чи використовується для цього оптимальний підхід. У рамках цього аналізу необхідно ретельно оцінити доступність та відповідність сучасних технологій, що потрібні для втілення

проєкту. Зважаючи на наявність декількох потенційних технічних рішень, кожен ключовий аспект проєкту має бути окремо проаналізований, з подальшим вибором найбільш релевантної технології для його реалізації.

В таблиці 6.3 представлені результати аудиту.

Таблиця 6.3 – Технологічна здійсненність ідеї проєкту

№	Ідея	Технологічні засоби реалізації	Наявність	Доступність
1	Мова програмування	C#	Наявний	Доступний
		Python	Наявний	Доступний
		C++	Наявний	Доступний
Обрана мова програмування для реалізації ідеї проєкта: Python				
2	Клієнтський інтерфейс для взаємодії	Angular	Наявний	Доступний
		React	Наявний	Доступний
		Streamlit	Наявний	Доступний
Обраний клієнтський інтерфейс для користувача: Streamlit				
3	Фреймворк для побудови архітектури нейронної мережі	Torch	Наявний	Доступний
		Tensorflow	Наявний	Доступний
Обраний тип послідовної мережі: Torch				
4	Тип мережі	Генеративно-змагальна	Наявний	Доступний
		На базі варіаційного автокодувальника	Наявний	Доступний
		Ймовірнісна модель дифузії	Власна розробка	Власна розробка
Обраний тип мережі: Ймовірнісна модель дифузії				

На підставі результатів проведеного аналізу можна дійти висновку, що всі обрані технологічні стеки є доступними, що підтверджує технічну здійсненність та готовність проєкту до впровадження.

6.3 Аналіз ринкових можливостей запуску стартап-проєкту

Наступним ключовим етапом є ідентифікація основних можливостей і потенційних загроз, які можуть виникнути під час реалізації проєкту.

Проведення такого аналізу дозволить глибше оцінити можливі ризики та розробити ефективні заходи для їх мінімізації. В таблиці 6.4 представлена попередня характеристика потенційного ринку стартап-проєкту.

Таблиця 6.4 – Попередня характеристика потенційного ринку стартап-проєкту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	$5 – 6 млрд
3	Динаміка ринку (якісна оцінка)	Легкий зріст
4	Наявність обмежень для входу (вказати характер обмежень)	Високі витрати на обчислювальні ресурси, сильна конкуренція, маркетингові витрати
5	Специфічні вимоги до стандартизації та сертифікації	Сертифікація безпеки даних, відповідність стандартам якості зображень та етичних принципів генерації
6	Середня норма рентабельності в галузі (або по ринку), %	20%–30%, залежно від сегменту (реклама, креативні індустрії, електронна комерція)

Згідно з результатами аналізу, ринок систем генерації зображень із текстових представлень характеризується наявністю обмеженої кількості основних гравців, які мають значні конкурентні переваги завдяки налагодженій базі клієнтів і потужній інфраструктурі. Помірний зріст ринку свідчить про сталий попит на подібні рішення, що дозволяє ключовим учасникам генерувати суттєві прибутки за рахунок рентабельності на рівні 20–30%. Це свідчить про перспективність галузі, особливо в контексті зростаючого попиту на автоматизацію творчих і бізнес-процесів.

Проте, слід враховувати певні бар'єри для входу на ринок, серед яких високі витрати на підтримку обчислювальних ресурсів, інтенсивна конкуренція та необхідність сертифікації. Сертифікація забезпечує відповідність стандартам безпеки даних і етичності, що є важливим фактором для стабільної роботи та довіри клієнтів.

Далі, визначимо вимоги потенційних клієнтів, розбивши їх на цільові групи відносно потреб, що формує ринок. Даний аналіз наведено в таблиці 6.5.

Таблиця 6.5 – Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Створення високоякісного візуального контенту	Рекламні агентства, маркетингові компанії, креативні студії	Високі вимоги до якості та відповідності зображень концепціям рекламних кампаній; важливість швидкості створення контенту	Максимальна деталізація та реалістичність зображень, швидкість обробки запитів, можливість кастомізації результату.

2	Автоматизація творчих процесів	Дизайнери, ілюстратори, розробники ігор	Потребують інтуїтивно зрозумілого інтерфейсу; важливий широкий вибір стилів та адаптивність до нестандартних запитів	Простота використання, адаптивність системи під запити, широкий вибір стилізацій, гарантія відповідності результатів очікуванням
3	Ефективна візуалізація освітніх та наукових концепцій	Освітні установи, наукові інститути, викладачі	Використання для створення візуалізацій складних тем; висока залежність від точності передавання наукових даних у зображеннях	Точність передавання даних, відповідність освітнім стандартам, безпека використання, стабільність роботи системи
4	Персоналізація цифрового контенту	Платформи електронної комерції, розробники NFT, особи, що створюють контент	Орієнтовані на індивідуальність та унікальність результатів; потребують можливості інтеграції з іншими платформами.	Унікальність зображень, інтеграція з іншими платформами, відповідність технічним стандартам, підтримка користувачів

Таким чином, основні сегменти цільової аудиторії для інтелектуальної системи генерації зображень із текстових представлень можна визначити як бізнес-компанії (рекламні агентства, платформи електронної комерції), освітні та наукові установи, а також приватні особи. Незважаючи на різноманітність потреб, такі як створення високоякісного візуального контенту, автоматизація творчих процесів

чи персоналізація цифрового контенту, вимоги до продукту залишаються подібними: надійність, точність, простота у використанні та можливість адаптації під специфічні запити клієнтів.

Отже, це створює потенціал для впровадження продукту на ринку.

Наступним етапом є оцінка факторів загроз та можливостей, які виникають при впровадженні проєкту, які представлені в таблицях 6.6 та 6.7.

Таблиця 6.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Високий рівень конкуренції	Значна присутність конкурентів на ринку, які пропонують схожі рішення з більш потужними ресурсами та досвідом.	Інвестування у диференціацію продукту, розробка унікальних функціональних можливостей, активна маркетингова кампанія.
2	Технологічні ризики	Можливі технічні збої у функціонуванні моделі, низька продуктивність або недоліки алгоритмів.	Регулярне тестування системи, залучення експертів для вдосконалення моделі, створення резервних технологічних рішень.
3	Етичні та правові обмеження	Можливі конфлікти, пов'язані з етичними питаннями або відповідністю законодавству щодо штучного інтелекту.	Розробка політики етичного використання продукту, дотримання міжнародних стандартів і сертифікацій.
4	Високі витрати на інфраструктуру	Значні фінансові затрати на обчислювальні потужності для роботи моделі.	Оптимізація алгоритмів для зниження ресурсомісткості, використання хмарних платформ для зменшення витрат.

Таблиця 6.7 – Фактори можливостей

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Зростання попиту на автоматизацію творчих процесів	Постійне зростання попиту на інструменти для автоматизації створення контенту у маркетингу, дизайні та освіті	Розширення функціональних можливостей системи, активне позиціонування продукту для різних сегментів ринку
2	Розвиток сучасних обчислювальних технологій	Інтеграція нових обчислювальних рішень (GPU, TPU) для підвищення швидкості та якості генерації	Інвестування у новітні технології, що підвищують ефективність роботи продукту, розробка партнерських відносин
3	Експоненціальний ріст використання штучного інтелекту	Постійне збільшення ролі ШІ у бізнес-процесах різних галузей	Адаптація продукту до нових сфер застосування, таких як медицина, архітектура чи персоналізація електронної комерції
4	Зростання інтересу до персоналізованого контенту	Користувачі шукають унікальні візуальні рішення для бізнесу чи персонального використання (NFT, брендинг)	Створення можливостей для гнучкої кастомізації продукту та інтеграції з платформами, які підтримують NFT-технології
5	Міжнародний ринок	Розширення географії продукту за рахунок попиту на автоматизовані системи візуалізації у різних країнах	Розробка багатомовного інтерфейсу, адаптація продукту до специфічних ринків, проведення міжнародних маркетингових кампаній

В результаті аналізу ринкового середовища виявлено низку загроз, які потребують реагування для забезпечення конкурентоспроможності проєкту. Серед ключових ризиків виділяються високий рівень конкуренції, значні технологічні виклики та можливі етичні та правові обмеження, що можуть вплинути на імідж та

функціональність системи. Разом із тим, вагомим бар'єром для входу є висока вартість інфраструктури, яка потребує оптимізаційних рішень для зниження ресурсомісткості. Для мінімізації цих ризиків необхідно зосередитися на інноваційності, етичності та підвищенні рівня обізнаності серед потенційних клієнтів.

З іншого боку, ринок відкриває значні можливості, серед яких зростання попиту на автоматизацію творчих процесів та персоналізований контент, розвиток обчислювальних технологій і розширення міжнародного впливу.

Далі, в таблиці 6.8 представлено ступеневий аналіз конкуренції на ринку.

Таблиця 6.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції: Монополістична	На ринку присутня значна кількість учасників, кожен із яких пропонує власні інноваційні рішення для генерації зображень із тексту, що створює середовище диференціації	Слід акцентувати увагу на унікальних можливостях продукту, таких як інноваційність алгоритмів, точність генерації та інтеграційні функції
Рівень конкурентної боротьби: Глобальний	Конкуренція здійснюється на міжнародному рівні, оскільки основні гравці, такі як OpenAI та Stability AI, працюють у різних регіонах світу	Необхідно розробити багатомовний інтерфейс, проводити адаптацію продукту до специфіки різних ринків та інвестувати у міжнародний маркетинг

Галузева ознака: Внутрішньогалузева	Конкуренція обмежена рамками однієї галузі, яка спеціалізується на системах штучного інтелекту для автоматизації візуалізації контенту	Зосередження на поглибленому розумінні галузі, постійна інновація продукту та партнерство із суміжними галузями для підсилення позицій на ринку
Конкуренція за видами товарів: Товарно-видова	Конкуренція виникає між різними моделями генерації зображень (DDPM, Latent Diffusion, GAN), які мають власні технічні переваги та обмеження	Розробка та впровадження сучасних дифузійних моделей, адаптація продукту до специфічних потреб клієнтів, акцент на точності та швидкості генерації
Характер конкурентних переваг: Нецінова	Головний акцент конкурентів робиться на якості зображень, інноваційності технологій, гнучкості системи та зручності інтерфейсу	Інвестування в технологічні розробки, підвищення якості сервісу, а також акцент на інтеграції системи з іншими платформами та персоналізації під користувача
Інтенсивність конкуренції: Марочна	Гравці на ринку активно просувають свої продукти як впізнавані бренди, орієнтуючись на унікальність та якість сервісу	Побудова сильної ідентичності бренду, розробка унікального позиціонування на ринку, активна участь у галузевих виставках, конференціях та рекламних кампаніях

Аналіз конкурентного середовища демонструє високий рівень інтенсивності конкуренції у межах монополістичного ринку з глобальним масштабом. Система генерації зображень із текстових описів стикається з викликами з боку численних гравців, які активно інвестують у технології, якість продуктів і розвиток бренду. У цих умовах підприємству необхідно зосередитися на технологічних інноваціях,

якості зображень, адаптивності системи до потреб клієнтів, а також на побудові сильної брендової ідентичності. Успіх залежить від здатності компанії виділити унікальні конкурентні переваги та ефективно їх комунікувати до цільової аудиторії.

Аналіз конкуренції за методологією М. Портера дозволяє здійснити всебічну оцінку взаємозв'язку між внутрішнім потенціалом організації та впливом ключових чинників зовнішнього середовища. Такий підхід сприяє визначенню оптимальної позиції компанії в галузі, що забезпечує максимальний рівень захисту від негативного впливу конкурентних сил або створює можливості для активного впливу на них з боку підприємства.

Результати аналізу конкурентного середовища за моделлю М. Портера наведені в таблиці 6.9.

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	DALL-E, MidJourney, Stable Diffusion	Малі студії або стартапи з новими моделями генерації зображень	Виробники GPU, хмарні сервіси для обчислень (AWS, GCP)	Графічні дизайнери, маркетологи, студії контенту	Фотографії, стокові зображення, програми 2D/3D графіки
Висновки:	Конкуренція висока через популярність сучасних моделей	Реальна можливість входу на ринок при інноваційному підході	Залежність від постачальників ресурсів для нейромереж	Клієнти вимагають високої якості, налаштувань та швидкості	Замінники можуть стати привабливими через доступність та простоту

Інтенсивність конкурентної боротьби у галузі є високою через наявність значних гравців, таких як DALL-E, MidJourney та Stable Diffusion, які вже зайняли значну частку ринку завдяки високій якості та широким функціональним можливостям своїх продуктів. Водночас існує реальна можливість входу на ринок

за умови впровадження інноваційних підходів, таких як створення унікальної моделі генерації зображень із новими функціями або адаптацією під потреби специфічних клієнтських сегментів. Постачальники, зокрема виробники GPU та хмарних обчислювальних платформ, диктують умови роботи через високу вартість ресурсів, що створює залежність від зовнішніх факторів. Клієнти мають високі вимоги до якості, швидкості та можливості налаштувань зображень, що зумовлює необхідність створення максимально зручного та ефективного інтерфейсу. Товари-замінники, такі як стокові зображення чи ручна робота, становлять певну загрозу, особливо через їхню доступність та низьку вартість, однак їхня обмеженість у швидкості створення та кастомізації залишає конкурентну перевагу за генеративними моделями. Таким чином, для успішної роботи на ринку необхідно створити продукт, який буде відрізнятися високою продуктивністю, зручністю використання та конкурентоспроможною ціною.

Оскільки наявність конкурентних переваг є важливою на даному ринку, виділено фактори конкурентоспроможності, які наведені у таблиці 6.10.

Таблиця 6.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1	Інноваційна архітектура генерації	Впровадження дифузійного механізму з адаптивною трансформацією на основі багаторівневої генерації, що дозволяє отримувати високоякісні результати навіть для складних вхідних умов. Унікальна архітектура моделі оптимізована для роботи з обмеженими обчислювальними ресурсами, що значно розширює її застосування.
2	Висока швидкість обробки	Завдяки інтеграції оптимізованих алгоритмів попередньої обробки та модульної структури нейромереж, система досягає значного зменшення затримок у процесі генерації. Використання багатопоточності у хмарному середовищі забезпечує розподіл задач для

		прискорення роботи навіть на великих наборах даних.
3	Універсальність застосування	Система адаптована до роботи з різними категоріями користувачів: від дизайнерів до великих компаній, завдяки широкому спектру можливостей налаштувань, включаючи формат виводу, деталізацію та стиль зображень.
4	Підвищена якість результатів	Завдяки постгенераційному модулю покращення зображень, система забезпечує оптимальну деталізацію, контрастність та кольорову гармонію, що дозволяє досягти результатів, які перевершують навіть традиційні методи створення графіки.
5	Мінімальні вимоги до обчислювальних ресурсів	Завдяки оптимізованій архітектурі моделі, система здатна працювати на пристроях із обмеженими обчислювальними ресурсами, такими як ноутбуки чи хмарні сервери з базовою конфігурацією. Це дозволяє значно розширити доступність технології для широкого кола користувачів.

На основі обґрунтування факторів конкурентоспроможності, проведено оцінку сильних та слабких сторін продукту. Порівняльний аналіз сильних та слабких сторін проєкту наведено в таблиці 6.11.

Таблиця 6.11 – Порівняльний аналіз сильних та слабких сторін проєкту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг систем конкурентів у порівнянні з розробленою системою						
			3	2	1	0	1	2	3
1	Інноваційна архітектура генерації	17						+	
2	Висока швидкість обробки	15				+			
3	Універсальність застосування	16					+		
4	Підвищена якість результатів	16					+		
5	Мінімальні вимоги до обчислювальних ресурсів	16					+		

Отже, результати порівняльного аналізу сильних і слабких сторін свідчать про наявність у системи ключових конкурентних переваг, які створюють сприятливі умови для її успішного просування на ринку.

Для комплексної оцінки поточної ситуації здійснюється SWOT-аналіз, який охоплює ринкові можливості, потенційні загрози, а також сильні та слабкі сторони проєкту. Результати SWOT-аналізу представлені в таблиці 6.12.

Таблиця 6.12 – SWOT- аналіз стартап-проєкту

Сильні сторони:	Слабкі сторони:
1. Розроблена технологія генерації забезпечує високу якість та адаптивність до різних вхідних параметрів. 2. Оптимізація обчислювальних ресурсів знижує витрати на інфраструктуру. 3. Універсальність функціоналу дозволяє задовольнити потреби різних категорій користувачів. 4. Швидкість генерації зображень забезпечує зменшення часу виконання завдань.	1. Висока залежність від хмарних сервісів та GPU для роботи з великими обсягами даних. 2. Відсутність розвиненої рекламної кампанії та низька впізнаваність бренду. 3. Значні витрати на підтримку інфраструктури та обслуговування. 4. Обмежена кількість попередньо налаштованих шаблонів у системі. 5. Недостатній досвід у взаємодії з широким колом ринкових сегментів.
Можливості:	Загрози:
1. Зростання попиту на автоматизовані інструменти створення графічного контенту. 2. Розширення галузей використання, зокрема у геймдеві, кіноіндустрії та освіті. 3. Розширення функціоналу через впровадження нових можливостей, таких як стилізація чи інтерактивні візуалізації. 4. Розвиток хмарних технологій забезпечує доступність обчислювальних потужностей для широкого використання.	1. Домінування на ринку сильних конкурентів, таких як DALL-E та MidJourney. 2. Швидкий розвиток нових технологій у конкурентів, які можуть перевершити існуючі рішення. 3. Висока чутливість клієнтів до ціни, що може обмежити попит на продукт. 4. Залежність від постачальників хмарних ресурсів, які можуть підвищити вартість своїх послуг.

Проведений SWOT-аналіз демонструє, що стартап-проєкт володіє суттєвими конкурентними перевагами, зокрема інноваційною архітектурою генерації, високою швидкістю роботи та універсальністю функціонального застосування, які забезпечують йому потенціал для ефективного виходу на ринок. Водночас аналіз виявив низку критичних слабких сторін, таких як залежність від зовнішніх постачальників обчислювальних ресурсів, недостатня впізнаваність бренду та обмеженість у попередньо налаштованих шаблонах, що може ускладнити завоювання ринкових сегментів. Ринкові можливості, зокрема зростання попиту на автоматизацію створення графічного контенту, інтеграція з іншими технологіями штучного інтелекту та розширення функціоналу продукту, відкривають перспективи для подальшого розвитку проєкту. Проте загрози, такі як висока конкуренція з боку провідних гравців, розвиток технологій у конкурентів, а також залежність від змін у регуляторному середовищі, створюють ризики, які потребують управління.

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки для виведення стартап-проєкту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проєкти конкурентів, що можуть бути виведені на ринок, які представлені в табл. 5.13.

Таблиця 6.13 – Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Проведення презентації технології для потенційних інвесторів	Висока	2 місяці
2	Підготовка мінімально життєздатного продукту (MVP) для демонстрації клієнтам	Середня	4 місяці
3	Інтеграція продукту з існуючими графічними платформами	Низька	6 місяців

4	Проведення рекламної кампанії для популяризації технології серед дизайнерів	Середня	5 місяців
5	Розробка комерційної версії продукту з розширеним функціоналом	Низька	8 місяців

На основі запропонованих альтернатив найдоцільнішим варіантом для початкового етапу впровадження стартап-проєкту є проведення презентації технології для потенційних інвесторів.

Ця альтернатива характеризується високою ймовірністю залучення ресурсів завдяки безпосередньому контакту з ключовими гравцями ринку, які можуть забезпечити фінансування подальших етапів розробки. Крім того, термін реалізації даного заходу становить лише 2 місяці, що дозволяє оперативно перейти до наступних етапів розвитку проєкту. Іншими перспективними варіантами є підготовка MVP для демонстрації клієнтам та проведення рекламної кампанії. Вони мають середню ймовірність отримання ресурсів і можуть сприяти формуванню базового інтересу з боку цільової аудиторії. Однак, ці заходи потребують більше часу на реалізацію (4-5 місяців), що може сповільнити подальше впровадження.

Альтернативи, пов’язані з інтеграцією продукту в екосистему графічних платформ і розробкою комерційної версії, є менш пріоритетними на початковому етапі, оскільки потребують значних інвестицій та мають нижчу ймовірність залучення ресурсів у короткі строки.

Їх реалізація є доцільною лише після успішного проходження етапу презентації технології та отримання початкового фінансування.

Таким чином, обраною альтернативою є проведення презентації технології для інвесторів як найбільш ефективний спосіб швидкого залучення ресурсів та підвищення впізнаваності продукту.

6.4 Розроблення ринкової стратегії проєкту

Розроблення ринкової стратегії передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 6.14).

Таблиця 6.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малий та середній бізнес (маркетингові агенції, дизайн-студії)	Висока: ці сегменти активно впроваджують інструменти автоматизації процесів	Високий: близько 50-60%	Висока	Середня
2	Великі корпорації (рекламні та медіа-компанії)	Помірна: попит залежить від інтеграції з існуючими системами	Середній: близько 40%	Висока	Низька
3	Приватні користувачі (дизайнери-фрилансери, блогери, контент-креатори)	Висока: продукт легко адаптується до індивідуальних потреб	Високий: близько 60-70%	Середня	Висока
4	Навчальні заклади та освітні платформи (університет, онлайн-курси)	Помірна: залежить від потреб у створенні навчального контенту	Середній: близько 30-40%	Середня	Середня
Які цільові групи обрано: Приватні користувачі (дизайнери, блогери)					

Основний акцент ринкової стратегії слід зробити на приватних користувачах, оскільки цей сегмент демонструє високу готовність прийняти продукт, значний потенційний попит та низькі бар'єри входу. Обрана група є ідеальною для початкового виходу на ринок і подальшого масштабування.

Для роботи з цим сегментом рекомендовано застосувати стратегію *концентрованого маркетингу,* що дозволить сфокусувати ресурси на задоволенні специфічних потреб цієї групи.

Далі, для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку, яка представлена в таблиці 6.15.

Таблиця 6.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Запуск мінімально життєздатного продукту (MVP) з базовим функціоналом	Концентрований маркетинг, орієнтований на приватних користувачів	Швидкість генерації, доступність, простий інтерфейс, кастомізація для індивідуальних потреб користувачів	Стратегія спеціалізації

Наступним кроком є вибір стратегії конкурентної поведінки, яка представлена в таблиці 6.16.

Таблиця 6.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару	Стратегія конкурентної поведінки*

			конкурента, і які?	
1	Ні	Так, з акцентом на залученні нових споживачів у нішевих сегментах	Ні, компанія зосередиться на унікальних функціях, таких як кастомізація стилів, швидкість генерації та простота інтеграції	Зайняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробляється стратегія позиціонування, що полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати торгівельну марку/проєкт, яка представлена в таблиці 6.17.

Таблиця 6.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту (три ключових)
1	Простий інтерфейс, інтуїтивно зрозумілий навіть для некваліфікованих користувачів	Спеціалізація	Легкість освоєння, доступність функціоналу для різних рівнів користувачів	Простота, доступність, універсальність
2	Швидкість генерації зображень та мінімальні затримки у роботі з великими обсягами даних	Спеціалізація	Висока продуктивність, оптимізовані алгоритми для швидкої обробки	Ефективність, продуктивність, надійність

3	Можливість налаштування параметрів генерації під індивідуальні потреби користувачів (розмір, стиль, кольори тощо)	Спеціалізація	Гнучкість налаштувань, персоналізація функцій під специфічні задачі	Адаптивність, інноваційність, кастомізація

Цільова аудиторія проєкту, зокрема приватні користувачі, висуває вимоги до продукту, які включають легкість у використанні, швидкість генерації контенту та можливість його адаптації під індивідуальні завдання.

Для задоволення цих потреб була обрана базова стратегія спеціалізації, яка дозволяє зосередитися на глибокому задоволенні потреб саме цієї групи споживачів, пропонуючи рішення, які не лише відповідають їхнім очікуванням, але й перевершують їх. Для формування ринкової позиції було обрано три ключові асоціації: інноваційність, яка підкреслює унікальність технології та сучасність підходу; простота, що демонструє легкість взаємодії з продуктом; та ефективність, яка акцентує увагу на якісному результаті.

6.5 Розроблення маркетингової програми стартап-проєкту

В таблиці 6.18 представлено визначення ключових переваг концепції потенційного товару

Таблиця 6.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Потреба у швидкому створенні якісного графічного контенту	Генерація зображень високої деталізації з мінімальними затримками завдяки оптимізованим алгоритмам	Швидкість і продуктивність обробки, які значно перевищують середні показники конкурентів

2	Потреба в адаптивності функціоналу для індивідуальних завдань	Налаштування параметрів генерації (розмір, стиль, кольорова гамма) дозволяють адаптувати продукт під конкретні потреби клієнта	Гнучкість та кастомізація, які роблять продукт універсальним для використання у різних галузях
3	Потреба в легкості використання продукту навіть для користувачів без технічного досвіду	Інтуїтивно зрозумілий інтерфейс та автоматизація основних процесів забезпечують легкість освоєння	Простота у використанні, яка мінімізує криву навчання для нових користувачів
4	Потреба у доступності технології за умов обмежених ресурсів	Продукт може працювати навіть на хмарних платформах з обмеженою потужністю, оптимізуючи використання обчислювальних ресурсів	Доступність для широкого сегменту клієнтів завдяки мінімальним технічним вимогам

Ключові потреби цільової аудиторії спрямовані на швидкість, якість, гнучкість та зручність у використанні, а також доступність навіть за обмежених ресурсів. Продукт стартапу надає користувачам можливість отримувати зображення високої якості за короткий час, зберігаючи при цьому легкість у роботі з системою завдяки інтуїтивному інтерфейсу. Гнучкість налаштувань відкриває перспективи для адаптації продукту до різних галузей, таких як маркетинг, графічний дизайн, кіноіндустрія та освіта. Вигоди, які забезпечує продукт, у поєднанні з конкурентними перевагами формують стійку концепцію, яка дозволяє ефективно залучати нових споживачів.

Далі, необхідно розробити трирівневу маркетингову модель товару: товар за задумом, товар в реальному часі та товар з підкріпленням (табл. 6.19).

Таблиця 6.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система автоматичної генерації графічного контенту із застосуванням дифузійних моделей, що забезпечує задоволення базової потреби користувачів у створенні високоякісних, адаптованих під індивідуальні потреби зображень. Основна функціональна вигода полягає у швидкому та зручному отриманні візуального контенту без залучення графічних дизайнерів або складних інструментів.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Швидкість роботи	Нм	Тх /Тл
	Гнучкість налаштувань	Нм	Тх /Тл
	Якість генерації	М	Тх /Тл
	Зручність використання	Нм	Тл
	Якість: відповідає стандартам ISO/IEC 25010, що гарантує надійність, зручність і продуктивність продукту.		
	Пакування: онлайн-сервіс із можливістю API-доступу та інтеграції у популярні графічні платформи		
	Марка: FastDDPM		
III. Товар із підкріпленням	До продажу: Функціональна система, яка включає можливість безкоштовного доступу до демо-версії для тестування функцій продукту.		
	Після продажу: Постійна технічна підтримка, оновлення програмного забезпечення, забезпечення безперебійного доступу до хмарної інфраструктури.		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності			

Наступним кроком є визначення цінових меж на розроблену систему, що представлено в таблиці 6.20.

Таблиця 6.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Стокові підписки від 10-15$ на місяць	Підписки на аналоги від 20-50$ на місяць	Середній (700-1500$ на місяць для приватних користувачів)	Нижня: 15$ на місяць, Верхня: 40$ на місяць

На наступному етапі визначається оптимальна систему збуту, яка представлена в в таблиці 6.21.

Таблиця 6.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнти очікують швидкий доступ до послуг і простоту придбання продукту онлайн	Забезпечення безперервного доступу до веб-платформи, підтримка API, регулярне оновлення	0-канал	Прямий збут через власну веб-платформу із можливістю інтеграції в партнерські сервіси

Фінальним етапом розробки маркетингової програми виступає формування концепції маркетингових комунікацій, яка базується на раніше визначених стратегічних принципах позиціонування та враховує особливості поведінкових характеристик цільової аудиторії, яка представлена в табл. 6.22.

Таблиця 6.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Клієнти орієнтовані на простоту використання продукту та швидке освоєння основних функцій	Соціальні мережі (Facebook, Instagram), таргетована реклама в пошукових системах	Простота, доступність, швидкість	Донести до потенційних клієнтів інформацію про зручність та легкість використання продукту	Практичні приклади взаємодії з інтерфейсом продукту, демонстрація швидкого результату

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
2	Професійні дизайнери та блогери прагнуть інноваційних інструментів для генерації контенту з унікальними характеристиками	Платформи для професійної аудиторії (LinkedIn, Behance), вебінари, онлайн-конференції	Інноваційність, кастомізація, якість	Показати, як унікальні можливості продукту сприяють реалізації професійних потреб	Візуальне порівняння результатів генерації, демонстрація адаптивних налаштувань
3	Споживачі очікують прозорості у ціноутворенні та доступності тарифних планів	Інтернет-сайт компанії, сторінки на маркетплейсах SaaS-продуктів (AppSumo, G2)	Доступність, гнучкість тарифів	Сформувати довіру до бренду шляхом підкреслення прозорості цінової політики	Інтерактивні калькулятори вартості, акцент на демо-доступ

Розроблена концепція базується на особливостях поведінки цільових клієнтів і охоплює ключові канали комунікацій, через які ефективно доноситься інформація про продукт. Основними завданнями є інформування клієнтів про простоту використання, інноваційність функціоналу, прозорість ціноутворення та технологічну надійність продукту.

Акцент робиться на практичну демонстрацію переваг через сучасні цифрові канали, які найбільш ефективно взаємодіють із обраною аудиторією.

Висновки до розділу 6

У шостому розділі проведено комплексний аналіз ринкових можливостей, технологічної здійсненності і стратегій виведення продукту на ринок, що

дозволило сформувати чітке бачення потенціалу стартап-проєкту. Основна ідея продукту полягає у створенні інноваційної системи генерації зображень на основі текстових описів із застосуванням дифузійних моделей.

Цей підхід дозволяє задовольнити широке коло потреб користувачів, забезпечуючи високу якість генерації, універсальність використання та простоту взаємодії із системою. Проведений технологічний аудит підтвердив здійсненність проєкту, обрано оптимальні технічні засоби, включаючи Python, Torch і Streamlit, які забезпечують продуктивність і доступність продукту навіть за умов обмежених обчислювальних ресурсів. Аналіз ринкових можливостей виявив стабільний попит на автоматизацію творчих процесів у таких галузях, як маркетинг, освіта, наука та розробка цифрового контенту. SWOT-аналіз продемонстрував, що ключовими перевагами продукту є інноваційна архітектура генерації, висока швидкість роботи та адаптивність системи, тоді як слабкими сторонами є низька впізнаваність бренду і обмеженість шаблонів. Для мінімізації ризиків запропоновано зосередитися на розробці унікальних функцій, інвестиціях у маркетинг і оптимізації інфраструктури. Основною цільовою аудиторією визначено приватних користувачів, таких як дизайнери, блогери та контент-креатори, що мають високу готовність сприймати продукт і значний потенціал попиту.

Вибрано стратегію концентрованого маркетингу з акцентом на інноваційність, простоту використання та швидкість роботи системи.

Розроблена трирівнева модель товару відображає чітку функціональність системи, передбачаючи швидку генерацію зображень, інтуїтивно зрозумілий інтерфейс і гнучкість налаштувань. Акцент зроблено на прозорості ціноутворення, демонстрації інноваційних можливостей продукту і його доступності. Отже, проведений комплексний аналіз підтверджує, що стартап-проєкт має значний потенціал для успішного виходу на ринок і створення конкурентоспроможного продукту, який задовольняє сучасні потреби користувачів у швидкому та якісному створенні графічних елементів.

ВИСНОВКИ

При роботі над даною магістерською дисертацією була розроблена інтелектуальна система генерації зображень на основі текстових описів, яка базується на ймовірнісних моделях дифузії.

Дослідження розпочалося з аналізу предметного середовища, в рамках якого було розглянуто сучасні генеративні моделі, їх архітектурні особливості, принципи роботи та можливості.

Основну увагу приділено таким моделям, як Imagen, Stable Diffusion і Runway ML, кожна з яких має свої унікальні переваги й обмеження. Це дозволило визначити актуальні підходи для розроблення власної системи.

На основі проведеного аналізу була сформульована постановка задачі, яка включала створення генеративної моделі для перетворення текстових даних у зображення, вибір архітектури системи, розробку програмного забезпечення для її реалізації та оцінку ринкового потенціалу продукту.

Далі, відповідно до структури дисертації, було виконано наступне:

– У другому розділі проведено огляд основних архітектур генеративних моделей, таких як генеративні змагальні мережі, варіаційні автокодувальники, моделі дифузії, авторегресивні моделі та нормуючі потоки. Особлива увага приділена дифузійним моделям через їх високу адаптивність.

– У третьому розділі описано математичні основи та технічні засоби, що забезпечують функціонування системи, включаючи використання бібліотек Torch, Diffusers, CLIP і TorchVision. Розроблено архітектуру системи генерації зображень із визначенням ключових компонентів, таких як попередня обробка даних, тренування моделі, генерація зображень і постгенераційна обробка.

– У четвертому розділі було реалізовано програмний цикл, включаючи підготовку вхідних даних, створення дифузійних перетворень, навчання нейронної мережі та оцінку продуктивності.

– У п’ятому розділі проведено тестування системи генерації зображень, яке підтвердило її функціональність, зручність використання та здатність точно

інтерпретувати текстові запити. Система забезпечує автентифікацію та реєстрацію користувачів, пропонує інтуїтивний інтерфейс для взаємодії та відображає прогрес генерації в реальному часі.

У заключному розділі було розроблено стартап-проєкт, що включав опис ідеї проєкту, технологічний аудит, аналіз ринкових можливостей, розробку ринкової стратегії та маркетингової програми. Було створено детальний план для успішного виведення системи на ринок із урахуванням конкурентного середовища та потреб потенційних клієнтів.

Таким чином, запропонована інтелектуальна система генерації зображень вирізняється високою якістю створюваних зображень, здатністю точно інтерпретувати текстові запити, універсальністю та зручністю використання. Її унікальність полягає в оптимізації процедури вибору часових кроків та інтеграції вдосконалених методів передбачення шуму, що зменшує кількість необхідних ітерацій для генерації зображення без втрати його якості.

Переваги запропонованої системи порівняно з існуючими рішеннями виявляються у високій продуктивності, спрощеній архітектурі та простоті інтеграції.

Загальний висновок дослідження свідчить про успішне розв'язання поставлених завдань у магістерській дисертації, а отже, мету дисертації було досягнуто.

Подальший розвиток системи передбачає комплексну роботу в кількох напрямах для забезпечення її масштабованості, продуктивності та відповідності сучасним вимогам користувачів. Розширення функціональних можливостей включає впровадження інтерфейсу для інтерактивного налаштування параметрів генерації, таких як стиль, композиція, кольорова гама та рівень деталізації зображень, що дозволить користувачам впливати на результати відповідно до їхніх індивідуальних потреб.

Також, планується впровадження механізмів використання оптимізованих обчислень для прискорення процедур генерації, а також оптимізованих бібліотек для мультипоточності й розподілених обчислень.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. O. Dalmaz, B. Saglam, G. Elmas, M. Mirza and T. Çukur. Denoising Diffusion Adversarial Models for Unconditional Medical Image Generation. 2023 *Signal Processing and Communications Applications Conference.* pp. 1-5. URL: https://ieeexplore.ieee.org/document/10223912 (дата звернення: 15.09.2024).

2. J. -U. Kim and D. -J. Kang. Enhancing Denoising Models Performance Through Timestep-Aware Predictor for Latent Diffusion-Based Image Generation. *International Conference on Automation and Systems.* pp. 1937-1940. URL: https://ieeexplore.ieee.org/document/10316977 (дата звернення: 16.09.2024).

3. W. Song et al. Medical Image Generation based on Latent Diffusion Models. *International Conference on Artificial Intelligence.* pp. 89-93. URL: https://ieeexplore.ieee.org/document/10497435 (дата звернення: 16.09.2024).

4. A. Iuliano, P. Liò, G. Manfredi and F. Romaniello. Denoising Probabilistic Diffusion Models for Synthetic Healthcare Image Generation. *IEEE International Workshop on Metrology for Living Environment.* 2024, pp. 415-420. URL: https://ieeexplore.ieee.org/document/10615511 (дата звернення: 16.09.2024).

5. Y. Kang, Z. Xie, L. Liu, X. Zhao and Y. Chen. Parameter-Guided Image Generation with Denoising Diffusion Models. 2023, pp. 4628-4633. URL: https://ieeexplore.ieee.org/document/10450463 (дата звернення: 16.09.2024).

6. M. Zhao, M. Liu, B. Ren, S. Dai and N. Sebe. Denoising Diffusion Probabilistic Models for Action-Conditioned 3D Motion Generation. *International Conference on Acoustics and Signal Processing.* 2024, pp. 4225-4229. URL: https://ieeexplore.ieee.org/document/10446185 (дата звернення: 22.09.2024).

7. C. Ham et al. Personalized Residuals for Concept-Driven Text-to-Image Generation. *Conference on Computer Vision and Pattern Recognition.* 2024, pp. 8186-8195. URL: https://ieeexplore.ieee.org/document/10658090 (дата звернення: 22.09.2024).

8. H. Ku, M. Lee, K. Ko and S. J. Baek .PoolImagen: Text-to-Image Diffusion Models With an Efficient Transformer Without Attention. *2024 International Conference*

on Artificial Intelligence in Information. 2024, pp. 123-128. URL: https://ieeexplore.ieee.org/document/10463464 (дата звернення: 22.09.2024).

9. H. Li, C. Shen, P. Torr, V. Tresp and J. Gu. Self-Discovering Interpretable Diffusion Latent Directions for Responsible Text-to-Image Generation. *IEEE Conference on Computer Vision and Pattern Recognition.* 2024, pp. 12006-12016. URL: https://ieeexplore.ieee.org/document/1067648 (дата звернення: 22.09.2024).

10. H. Nam, J. Park, J. Choi and S. -L. Kim. Sequential Semantic Generative Communication for Progressive Text-to-Image Generation. *IEEE International Conference on Communication, and Networking*. 2023, pp. 91-94. URL: https://ieeexplore.ieee.org/document/10287475 (дата звернення: 22.09.2024).

11. B. P. Prathaban, R. Subash, A. Ashwini, D. F. D. Shahila and R. Prasanna. VQGAN Text-to-Image Generation Enhancement using CLIP. International Conference on Communication, Computing and IoT. 2024, pp. 1-6. URL: https://ieeexplore.ieee.org/document/10550382 (дата звернення: 26.09.2024).

12. B. Puranik, R. Inamdar, P. Ghule, S. Awlankar and P. Ovhal. Insights and Performance Evaluation of Stable Diffusion Models with OpenVINO on Intel Processor. *3rd World Conference on Applied Intelligence and Computing.* 2024, pp. 1408-1411. URL: https://ieeexplore.ieee.org/document/10731090 (дата звернення: 26.09.2024).

13. W. Cai, J. Xiang and B. Hu. Self-Modulated Feature Fusion GAN for Text-to-Image Synthesis. *International Conference on Engineering and Emerging Technologies*. 2023, pp. 1-6. URL: https://ieeexplore.ieee.org/document/10526149 (дата звернення: 26.09.2024).

14. Denoising Diffusion Probabilistic Models. GithubPages : офіційний сайт. URL: https://hojonathanho.github.io/diffusion/ (дата звернення: 26.09.2024).

15. H. P. Thethi, A. Rajitha, V. Revathi, R. Chourasia, D. K. Yadav and M. N. Abed. Integrating the Innovations of GPT-3 and U-Net for Text-to-Image Creation in Digital Art Galleries. *International Technology Conference on Smart Computing for Innovation and Advancement in Industry 4.0.* 2024, pp. 1-6. URL: https://ieeexplore.ieee.org/document/10687894 (дата звернення: 30.09.2024).

16. O. Zambrano and B. Senouci. Image Classification Improvement: Text-to-Image AI for Synthetic Dataset Approach. *Euromicro Conference on Software Engineering and Advanced Applications.* 2023, pp. 74-77. URL: https://ieeexplore.ieee.org/document/10371557 (дата звернення: 05.10.2024).

17. K. Sueyoshi and T. Matsubara. Predicated Diffusion: Predicate Logic-Based Attention Guidance for Text-to-Image Diffusion Models. *IEEE Conference on Computer Vision and Pattern Recognition.* 2024, pp. 8651-8660. URL: https://ieeexplore.ieee.org/document/10657728 (дата звернення: 05.10.2024).

18. S. S. Baraheem and T. V. Nguyen. Aesthetic-Aware Text to Image Synthesis. Annual Conference on Information Sciences and Systems 2020, pp. 1-6. URL: https://ieeexplore.ieee.org/document/9086262 (дата звернення: 05.10.2024).

19. S. Jayasumana, D. Glasner, S. Ramalingam, A. Veit, A. Chakrabarti and S. Kumar. MarkovGen: Structured Prediction for Efficient Text-to-Image Generation. *IEEE Conference on Computer Vision and Pattern Recognition.* 2024, pp. 9316-9325. URL: https://ieeexplore.ieee.org/document/10656397 (дата звернення: 12.10.2024).

20. L. Xiaolin and G. Yuwei. Research on Text to Image Based on Generative Adversarial Network. *International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China.* 2020, pp. 330-334. URL: https://ieeexplore.ieee.org/document/9422034 (дата звернення: 14.10.2024).

21. L. Qu, W. Wang, Y. Li, H. Zhang, L. Nie and T. -S. Chua. Discriminative Probing and Tuning for Text-to-Image Generation. IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, pp. 7434-7444. URL: https://ieeexplore.ieee.org/document/10655389 (дата звернення: 14.10.2024).

22. P. Chávez and W. Ticona, "Implementation of Text-to-Image Generators in the Development of the Usability Interface for the Construction of a Web Page. *International Conference on Cloud Computing, Data Science & Engineering.* 2024, pp. 926-930. URL: https://ieeexplore.ieee.org/document/10463436 (дата звернення: 14.10.2024).

23. J. Mao and X. Wang. Training-Free Location-Aware Text-to-Image Synthesis. International Conference on Image Processing. 2023, pp. 995-999. URL: https://ieeexplore.ieee.org/document/10222616 (дата звернення: 14.10.2024).

24. A. MaungMaung, H. H. Nguyen, H. Kiya and I. Echizen. Fine-Tuning Text-To-Image Diffusion Models for Class-Wise Spurious Feature Generation," IEEE International Conference on Image Processing. 2024, pp. 3910-3916. URL: https://ieeexplore.ieee.org/document/10647627 (дата звернення: 14.10.2024).

25. M. Rohith, L. Pallavi, K. Shirisha, M. Sanjay and V. S. Priya. Image Generation Based on Text Using BERT And GAN Model. International Conference on Computational Intelligence Networking. 2023, pp. 214-218. URL: https://ieeexplore.ieee.org/document/10141495 (дата звернення: 14.10.2024).

26. Z. Wang et al., Generative Adversarial Networks Based on Dynamic Word-Level Update for Text-to-Image Synthesis. International Conference on Image, Vision and Computing (ICIVC). 2022, pp. 641-647. URL: https://ieeexplore.ieee.org/document/9886095 (дата звернення: 14.10.2024).

27. S. Mahajan, T. Rahman, K. M. Yi and L. Sigal. Prompting Hard or Hardly Prompting: Prompt Inversion for Text-to-Image Diffusion Models. IEE Conference on Computer Vision and Pattern Recognition. 2024, pp. 6808-6817. URL: https://ieeexplore.ieee.org/document/10658265 (дата звернення: 20.10.2024).

28. M. F. Sutedy and N. N. Qomariyah. Text to Image Latent Diffusion Model with Dreambooth Fine Tuning for Automobile Image Generation. International Seminar on Research of Information Technology. 2022, pp. 440-445. URL: https://ieeexplore.ieee.org/document/10052908 (дата звернення: 20.10.2024).

29. CLIP: Connecting text and images. OpenAI: офіційний сайт. URL: https://openai.com/index/clip/ (дата звернення: 20.10.2024).

30. ImageNet dataset. URL: https://paperswithcode.com/dataset/imagenet (дата звернення: 20.10.2024).

31. Radford, A., Kim, J. W., Hallacy, C., Sastry, G., Askell, A. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv preprint arXiv:2103. URL: https://arxiv.org/abs/2103.00020 (дата звернення: 26.10.2024).