Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

ПРИКЛАДНІ ЗАСТОСУВАННЯ СУЧАСНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ СИНТЕЗУ ЗОБРАЖЕНЬ
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-20-3

Бєлєвцева А.Р.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Гороховатський В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Бєлєвцевій Анастасії Русланівні
(прізвище, ім'я, по батькові)

1. Тема роботи Прикладні застосування сучасних нейронних мереж для синтезу зображень

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 01 червня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, засоби комп'ютерної графіки, аналіз застосувань сучасних нейромереж в сфері синтезу зображень.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз предметної області та постановка задачі.
2. Математичне обґрунтування вибраних застосувань нейромереж.
3. Нейронні мережі у системах комп'ютерної графіки.
4. Реалізація прикладів задач із використанням нейромереж.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми синтезу зображень, постановка задачі, огляд та аналіз архітектур, розгляд прикладів використання нейронних мереж у різних галузях, переваги та недоліки використання нейронних мереж, визначення сучасних застосувань синтезу зображень.

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.04.2024	
2	Аналіз завдання, підбір літератури	08.04.24-10.04.24	
3	Аналіз літератури з досліджуваної проблеми	11.04.24-20.04.24	
4	Аналіз предметної області	21.04.24-01.05.24	
5	Розробка методів	02.05.24-04.05.24	
6	Програмна постановки задачі	05.05.24-09.05.24	
7	Оформлення пояснювальної записки	10.05.24-30.05.24	
8	Перевірка на плагіат	01.06.24	
9	Рецензування	02.06.24	
10	Підготовка презентації та доповіді	29.05.24-02.06.24	
11	Занесення роботи в електронний архів	10.06.24	
12	Попередній захист кваліфікаційної роботи	10.06.24	

Дата видачі завдання 8 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ ___проф. Гороховатський В.О.___
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 63 с., 18 рис., 50 джерел.

НЕЙРОСИНТЕЗ, ШТУЧНИЙ ІНТЕЛЕКТ, ТРАНСФОРМАЦІЯ ЗОБРАЖЕНЬ, GANs, ПЕРЕНОС СТИЛЮ, ГЕНЕРАЦІЯ ЗОБРАЖЕНЬ.

Об’єктом роботи є нейронна мережа *DeepAI*, здатна до синтезу зображень, в той час як предметом є процес генерації високоякісних візуальних контентів.

Мета даної роботи полягає у ретельному вивченні найновіших методів синтезу зображень нейронної мережі – *DeepAI*, що включає в себе такі методи як: генерацію зображень на основі тексту (Text-to-Image Synthesis), перенос стилю (Style Transfer), генеративні протилежні мережі (GANs).

Синтез зображень за допомогою нейронних мереж є актуальним та перспективним напрямком новітніх розробок у галузі штучного інтелекту та комп’ютерної графіки.

У роботі проведено аналіз моделей нейронних мереж для синтезу зображень, розглянуто підходи до їх оптимізації, зокрема Adam, методи переносу стилю і генерації зображень на основі тексту. Особливу увагу приділено прикладним застосуванням нейронних мереж у графічному дизайні та з акцентом на технології *DeepAI*.

NEUROSYNTHESIS, ARTIFICIAL INTELLIGENCE, IMAGE TRANSFORMATION, GANs, STYLE TRANSFER, IMAGE GENERATION.

The object of this work is the *DeepAI* neural network capable of image synthesis, while the subject is the process of generating high-quality visual content.

The purpose of this work is to thoroughly study the latest methods of image synthesis of the neural network – *DeepAI*, which includes such methods as: text-based image generation (Text-to-Image Synthesis), Style Transfer, and Generative Adversarial Networks (GANs).

Image synthesis using neural networks is a relevant and promising area of recent developments in the field of artificial intelligence and computer graphics.

The paper analyzes neural network models for image synthesis, considers approaches to their optimization, including Adam, style transfer methods, and text-based image generation. Particular attention is paid to the applications of neural networks in graphic design with an emphasis on *DeepAI* technology.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 6
Вступ 7
1 Аналіз предметної області та постановка задачі 9
1.1 Огляд сучасних нейромережевих засобів синтезу зображень 9
1.2 Прикладні застосування нейронних мереж у сфері синтезу зображень 14
1.3 Аналіз сильних та слабких сторін методів синтезу зображень 22
1.4 Постановка задач роботи 26
2 Елементи моделювання і програмування нейронних мереж 27
2.1 Аналіз моделей нейронних мереж 27
2.2 Навчання мереж для синтезу зображень 31
2.3 Аналіз збіжності та стабільності методів 38
3 Реалізація программної моделі та тестування 44
3.1 Опис розробленого програмного забезпечення для синтезу зображень 44
3.2 Проведення експериментів та тестування нейронної мережі 47
3.3 Аналіз отриманих результатів 53
Висновки 57
Перелік джерел посилання 58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GANs – Generative Adversarial Network (генеративні протилежні мережі)

GPU – Graphics Processing Unit (графічних процесорів)

CNN – Cable News Network (нейронні мережі)

RNN – Recurrent Neural Network (рекурентні нейронні мережі)

SGD – Stochastic Gradient Descent (стохастичний градієнтний спуск)

Adam – Adaptive Moment Estimation (адаптивні оптимізаційні методи)

SSIM – Structural Similarity Index (індекс структурної схожості)

MNIST – Modified National Institute of Standards and Technology (об’ємна база даних рукописних зразків цифр)

PSNR – Peak Signal-to-Noise Ratio (пікове співвідношення сигналу до шуму)

ВСТУП

Синтез зображень за допомогою впровадження нейронних мереж став однією з найбільш обговорюваних та перспективних тем в сфері машинного навчання та комп’ютерного зору [1-6]. Завдяки значним досягненням в глибокому навчанні, нейронні мережі здобули визнання як потужний інструмент для генерації високоякісних зображень, що має великий потенціал у різних сферах, включаючи медицину, дизайн, ігрову індустрію, та інші [2-9].

З розвитком технологій глибокого навчання і появою потужних алгоритмів, таких як генеративні протилежні мережі (GANs), нейронні мережі стали здатні генерувати складні та реалістичні зображення, які важко відрізнити від реальних. GANs складаються з двох основних моделей: генератора, який створює зображення, і дискримінатора, який оцінює їх якість [4]. Це відкриває нові можливості для створення контенту, редагування зображень, візуальних ефектів у кіно та іграх, а також для розв’язання завдань у медичній діагностиці та реконструкції [10-14].

У роботі буде використано літературний огляд, експериментальне моделювання та проведено аналіз результатів. Будуть застосовані методи машинного навчання, статистичного аналізу та візуальної оцінки згенерованих зображень.

Практична цінність полягає в можливості використання отриманих знань для розробки нових технологій та застосувань у практиці.

Розвиток нейронних мереж, зокрема генеративних протилежних мереж (GANs), розширює можливості створення складних та візуально реалістичних зображень. Вивчення методів тренування таких як стохастичний градієнтний спуск (SGD), адаптивні методи оптимізації, наприклад, Adam або RMSprop та оптимізації нейронних мереж є важливим етапом для досягнення заданих результатів у генерації зображень. Аналіз основних архітектур нейронних мереж дозволяє виявити оптимальні підходи до синтезу зображень з високою якістю. Оцінка якості та реалістичності згенерованих зображень відіграє

ключову роль у визначенні ефективності та придатності нейросинтезу для різних цілей [15-19].

Загалом, синтез зображень за допомогою нейронних мереж є актуальною та перспективною темою, яка привертає все більше уваги дослідників та фахівців у галузі штучного інтелекту та комп’ютерної графіки [20-23]. Ця робота спрямована на розширення розуміння можливостей цієї технології, нейронні мережі, такі як *DeepAI*, виявляються ефективним інструментом для створення високоякісного контенту в різних галузях, включаючи кінематограф та медицину та її застосування для вирішення реальних прикладних завдань, таких як реставрація та покращення якості зображень, медична діагностика та інше [24-28].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд сучасних нейромережевих засобів синтезу зображень

Синтез зображень представляє собою ключовий аспект комп'ютерної графіки та штучного інтелекту, фокусуючись на розробці алгоритмів, які можуть створювати візуальний контент з нуля або на основі існуючих даних. Цей процес є критичним для багатьох застосувань, включаючи візуальні ефекти у кіноіндустрії, комп'ютерну графіку, медичну діагностику та наукові дослідження.

Історичний огляд розвитку методів синтезу зображень відображає значущі етапи та технологічні досягнення, що сприяли еволюції цієї області науки і техніки [22]. З перших експериментів у графічному програмуванні до сучасних алгоритмів машинного навчання синтез зображень пройшов довгий шлях від простих растрових зображень до складних інтерактивних візуальних сцен за допомогою застосуванням новітніх технологій *DeepAI*.

Початкові етапи розвитку методів синтезу зображень: генеративні моделі на основі автоенкодерів, варіанти опису об'єктів та сцен, перші застосування глибокого навчання**,** були періодом експериментування та відкриття в галузі комп'ютерної графіки. Від самого початку, коли комп'ютери стали доступними для широкого кола користувачів, виникла потреба в інструментах для генерації візуального контенту [29-33].

Художники та дизайнери в ранні періоди комп'ютерної графіки стикалися з викликом ручного моделювання та анімації візуальних елементів. Цей процес вимагав значних зусиль та часу, і результати часто обмежувалися простими формами та текстурами через обмежені обчислювальні ресурси.

З розвитком технологій у 1980-х роках з'явилися перші програмні інструменти для автоматизації генерації зображень. Ці ранні системи рендерингу дозволили створювати більш складні та деталізовані візуальні

ефекти, ніж це було можливо раніше. Вони відкрили нові можливості для створення фотореалістичних зображень та анімацій, що революціонізували індустрію кіно та відеоігор.

У 1990-2000-х роках настала нова ера в розвитку методів синтезу зображень, завдяки значним технологічним проривам в області графічних процесорів та алгоритмів рендерингу. Цей період був вирішальним для прогресу в комп'ютерній графіці, який поклав основу для сучасних стандартів реалістичності та продуктивності [20].

Прикладом сучасних досліджень є використання *DeepAI* для створення візуальних ефектів у фільмах, де ця технологія дозволяє створювати складні анімації та спецефекти. Крім того, в комп'ютерних іграх *DeepAI* застосовується для генерації текстур, моделювання середовищ і створення персонажів, що поліпшує імерсивний досвід гравців. У медичній галузі *DeepAI* допомагає покращити точність обробки медичних зображень, таких як рентгенівські знімки або томографія, що сприяє швидшому виявленню патологій та захворювань, приклад роботи технології показано на рисунку 1.1.

а)

б)

Рисунок 1.1 – Приклад роботи обробки *DeepAI*:
а) зображення рентгенівського знімку до обробки; б) зображення рентгенівського знімку після оброки *DeepAI* – Super Resolution

Розвиток графічних процесорів (GPU) дозволив значно підвищити обчислювальну потужність та швидкість рендерингу, що робило можливим обробку складних графічних сцен у реальному часі. Це відкрило нові горизонти для створення візуально захоплюючих ігор, фільмів та інтерактивних додатків з фотореалістичною графікою та високим рівнем деталізації [16].

Алгоритми рендерингу також пройшли значний еволюційний процес, що дозволив досягти більшої точності, ефективності та реалістичності в синтезі зображень. Рендеринг у реальному часі став стандартом для багатьох застосувань, включаючи відеоігри та візуалізації, де висока швидкість обробки інтерактивних сцен є критичною [8].

Генеративні протилежні мережі (GANs), метод яких використовує *DeepAI*, стали однією з ключових інновацій, які дозволили генерувати високоякісні та реалістичні зображення. Ці нейронні мережі, складаючись з генератора та дискримінатора, здатні створювати візуальний контент, який важко відрізнити від справжніх фотографій, відкриваючи нові горизонти для графічного дизайну, медіа-арту та візуальних ефектів у кіно.

Автоенкодери, з іншого боку, використовуються для стиснення та відновлення зображень, забезпечуючи ефективний метод для реконструкції та модифікації візуальних даних. Вони дозволяють не лише відтворювати, але й трансформувати зображення, створюючи нові творчі варіанти на основі існуючих даних [5-7].

Трансформери, спочатку розроблені для обробки послідовностей в області природної мови, також виявилися ефективними у генерації зображень, особливо при роботі з текстовими описами. Їхні механізми уваги дозволяють адаптуватися до різноманітних стилів та контекстів, створюючи деталізовані та змістовні візуальні представлення на основі тексту [12].

Генерація фотореалістичних зображень є однією з основних цілей сучасного синтезу зображень. Це вимагає розробки алгоритмів, які можуть відтворювати складні деталі, текстури та освітлення, що є характерними для

реального світу. Генеративні протилежні мережі (GANs) стали основним інструментом у цій сфері, здатними виробляти високоякісні та реалістичні зображення за допомогою складних архітектур нейронних мереж.

Крім того, синтез зображень також включає в себе створення абстрактних візуальних елементів, які можуть бути використані для творчих експериментів та візуального мистецтва. Трансформери, які спочатку були розроблені для обробки послідовностей даних, виявилися ефективними у генерації деталізованих та комплексних зображень, які можуть включати різноманітні стилі, форми та кольорові палітри [25].

Генеративні мережі (GANs) є потужним класом нейронних мереж, який здобув велику популярність завдяки їх здатності генерувати високоякісні та реалістичні зображення. Структура GANs базується на концепції двох конкуруючих нейронних мереж: генератора і дискримінатора [16].

Генератор відповідає за створення зображень. Він приймає на вхід випадковий шум або інші форми даних і генерує зображення, яке, за попередніми критеріями, має бути якнайбільш схожим на справжні. Генератор працює на основі послідовних трансформацій цього вхідного шуму, перетворюючи його в складні структури, які відповідають характеристикам дійсних зображень [18].

Дискримінатор, з іншого боку, діє як суддя, який оцінює створені генератором зображення. Його основне завдання – визначити, чи є зображення, представлене генератором, реалістичним, тобто чи воно схоже на справжні зображення з навчального набору даних. Дискримінатор отримує на вхід як справжні зображення, так і згенеровані генератором, і намагається відрізнити одне від іншого [30].

Процес навчання GANs базується на конкурентній динаміці між генератором і дискримінатором. Генератор намагається підвищити якість своїх зображень, щоб обдурити дискримінатора, тоді як дискримінатор намагається вдосконалити свої навички розрізнення між справжніми та згенерованими зображеннями. Ця конкурентна динаміка сприяє покращенню

якості генерованих зображень з кожним кроком навчання, що призводить до виробництва високоякісних і реалістичних візуальних контентів [29].

Автоенкодери є важливим класом нейронних мереж, які використовуються в синтезі зображень, реконструкції та компресії даних. Структурно автоенкодери складаються з двох основних компонентів: енкодера і декодера [31].

Енкодер приймає на вхід зображення і перетворює його в компактний вектор або код, який представляє суттєві характеристики зображення. Цей код є результатом стиснення інформації з вхідного зображення в менший простір, зберігаючи при цьому ключові особливості і структуру даних [29-34].

Після отримання коду декодер виконує обернений процес, приймаючи код і відновлюючи оригінальне зображення з цього компактного представлення. Метою автоенкодера є навчитися ефективно стиснувати і відновлювати дані, зберігаючи при цьому якість і деталі зображення [32].

Автоенкодери, можуть бути використані в різних задачах, таких як реконструкція зображень після їхньої компресії, а також для генерації нових зображень, які мають схожі характеристики з навчального набору даних. Вони базуються на автоенкодерах, але включають ймовірнісну модель, що дозволяє їм генерувати більш різноманітні і якісні зображення. Завдяки їхній здатності вчитися відтворювати інформацію, автоенкодери можуть бути ефективними інструментами для генерації високоякісних зображень та візуальних ефектів, особливо коли недоступний великий об’єм анотованих даних [21].

Трансформери, ініційовані для обробки послідовностей в контексті природної мови, здобули значний успіх завдяки своїй здатності використовувати механізми уваги для аналізу довгих залежностей в даних. Ця архітектура виявилася не лише ефективною для текстових даних, але й демонструє великий потенціал у задачах синтезу зображень [9].

Трансформери можуть бути адаптовані для генерації зображень, зокрема для перетворення текстових описів на візуальний контент. Цей процес включає в себе створення зображення, яке відповідає текстовому опису (Text-

to-Image Synthesis), і може бути особливо корисним у задачах, де необхідно автоматично генерувати візуальний контент на основі текстової інформації [9-11].

Мережі з умовною генерацією (Conditional GANs) є потужними інструментами у галузі генеративного моделювання, які розширюють базову ідею генеративних протилежних мереж (GANs), додавши концепцію умови або контексту у процес створення зображень. Вони роблять процес генерації зображень більш гнучким і керованим залежно від введених умов [14].

1.2 Прикладні застосування нейронних мереж у сфері синтезу зображень

Прикладні застосування нейронних мереж у сфері синтезу з *DeepAI* зображень демонструють широкий спектр можливостей, які відкриваються завдяки сучасним технологіям глибокого навчання. Нейронні мережі, такі як генеративні протилежні мережі (GANs), автоенкодери та трансформери, забезпечують ефективні та гнучкі рішення для генерації візуального контенту. *DeepAI* використовується для створення високоякісних зображень з використанням глибоких генеративних моделей, таких як генеративні протилежні мережі (GANs), які дозволяють генерувати фотореалістичні зображення з нуля, приклад роботи показано на рисунку 1.2.

Рисунок 1.2 – Зображення, яке було згенеровано за допомогою технологій *DeepAI*

Одним з ключових застосувань є Text-to-Image Synthesis, де нейронні мережі перетворюють текстовий опис на візуальні зображення [35]. Це дозволяє автоматизувати процес створення візуальних контентів для різних застосувань, включаючи медіа-індустрію, креативні проєкти та дизайн (рис. 1.3).

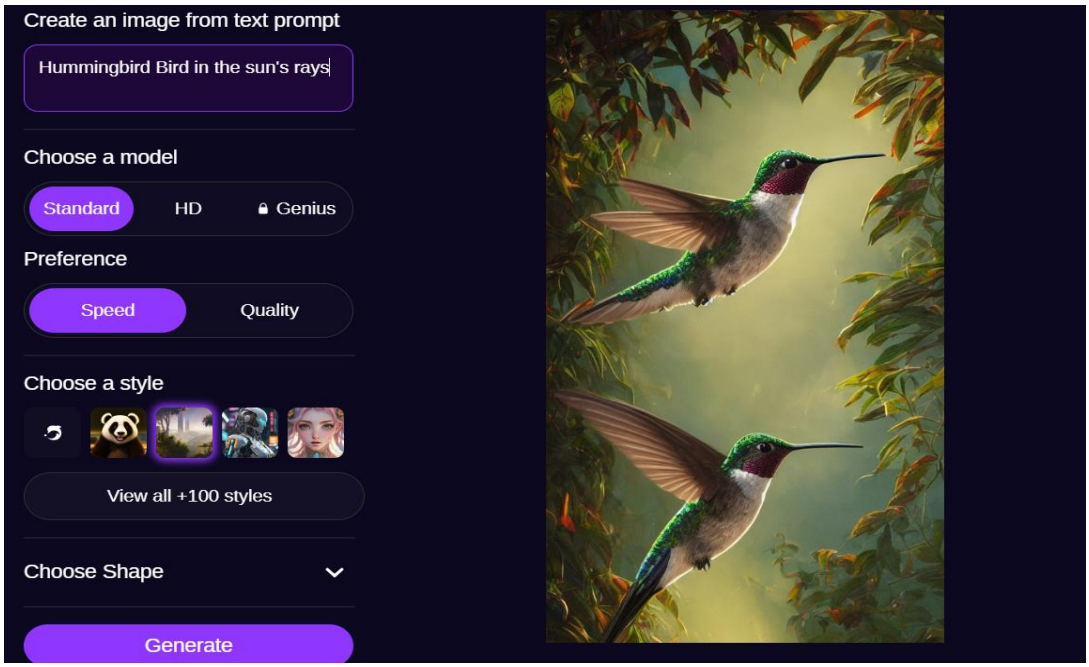

Рисунок 1.3 – Генерація зображення з застосуванням Text-to-Image Synthesis у *DeepAI*

Також *DeepAI* активно використовується для Style Transfer, техніки, що дозволяє переносити стиль одного зображення на інше, зберігаючи при цьому основні вмістові аспекти. Це створює можливості для індивідуалізації та кастомізації візуальних матеріалів у різних галузях, включаючи мистецтво, рекламу та вебдизайн (рис. 1.4) [25-29].

Нейронні мережі в області графічного дизайну та обробці зображень відіграють ключову роль у створенні новаторських та унікальних візуальних робіт. Одним із найцікавіших застосувань цих технологій є стильове перенесення (Style Transfer), яке дозволяє адаптувати художній стиль одного зображення до іншого, формуючи зовсім нове і вражаюче візуальне вираження [34-41]. Цей процес базується на здатності нейронних мереж розпізнавати та

адаптувати характеристики стилю, такі як текстури, колірні палітри та форми, з одного зображення на інше. Результат може бути дивовижним поєднанням елементів реалізму та художнього вираження, створюючи унікальні візуальні ефекти та арт-роботи [42].

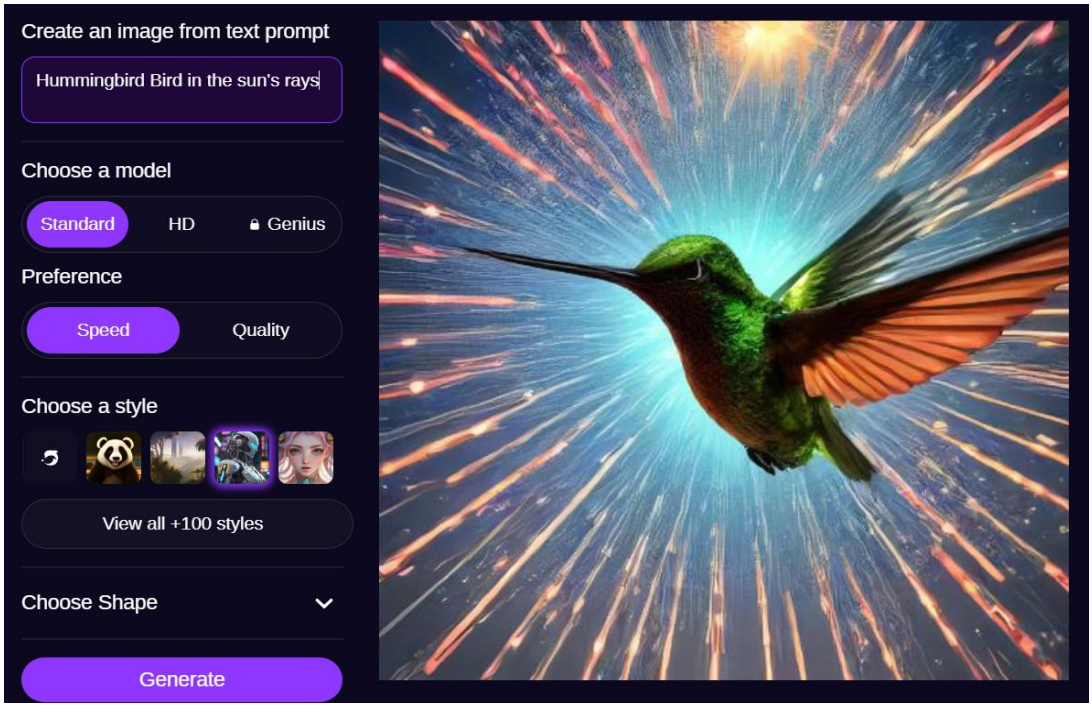

Рисунок 1.4 – Генерація зображення з застосуванням іншого Style Transfer у *DeepAI*

Ця техніка відкриває безліч можливостей для творчості в сфері графічного дизайну, мистецтва та культурного вираження. Вона може бути використана для створення інноваційних арт-проєктів, рекламних кампаній, графічних дизайнів для вебсайтів, книжкового дизайну та багатьох інших областей, де креативність та візуальний вплив мають вирішальне значення [30-35].

Генеративні протилежні мережі (GANs) складаються з двох основних компонентів: генератора і дискримінатора. Генератор відповідає за створення зображень, тоді як дискримінатор намагається відрізнити справжні зображення від згенерованих. Ця взаємодія стимулює навчання генератора генерувати все більш реалістичні та деталізовані зображення, що відкриває

безліч можливостей для синтезу фотореалістичних зображень, анімацій та візуальних ефектів у реальному часі.

Автоенкодери, в свою чергу, використовуються для стиснення та відновлення зображень, забезпечуючи ефективний метод для реконструкції та модифікації візуальних даних. Їх архітектура складається з енкодера, який перетворює вхідне зображення в складний вектор-код, та декодера, який відновлює оригінальне зображення з цього коду. Це дозволяє не лише відтворювати, але й трансформувати зображення, створюючи нові творчі варіанти на основі існуючих даних [9].

Трансформери, які спочатку були розроблені для обробки послідовностей в області природної мови, також виявилися дуже ефективними у генерації зображень. Їхні механізми уваги дозволяють адаптуватися до різноманітних стилів та контекстів, створюючи деталізовані та змістовні візуальні представлення на основі текстових описів [17].

Нейронні мережі знайшли широке застосування в графічних та обробці зображень, демонструючи свою ефективність та гнучкість в різних областях. Ось декілька конкретних прикладів застосування [1].

Генеративні протилежні мережі (GANs) є потужними інструментами в області графічного дизайну та синтезу зображень. Їх унікальна архітектура, яка складається з двох компонентів – генератора та дискримінатора, дозволяє досягати вражаючої якості та реалістичності у створенні нових візуальних контентів [2]. Наприклад, GANs можуть бути використані для генерації фотореалістичних зображень у сфері реклами та маркетингу, а також для відновлення та реставрації старих або пошкоджених фотографій, що має велике значення для архівів та музеїв (рис. 1.5).

Генератор в GANs функціонує як художник або дизайнер, який намагається створити зображення з нуля, під час як дискримінатор використовується як критик, що оцінює, наскільки добре генератор виконав свою роботу. Цей постійний цикл зворотного зв’язку підтримує процес

навчання, допомагаючи генератору вдосконалювати свої навички та створювати все більш якісні та реалістичні зображення [13-15].

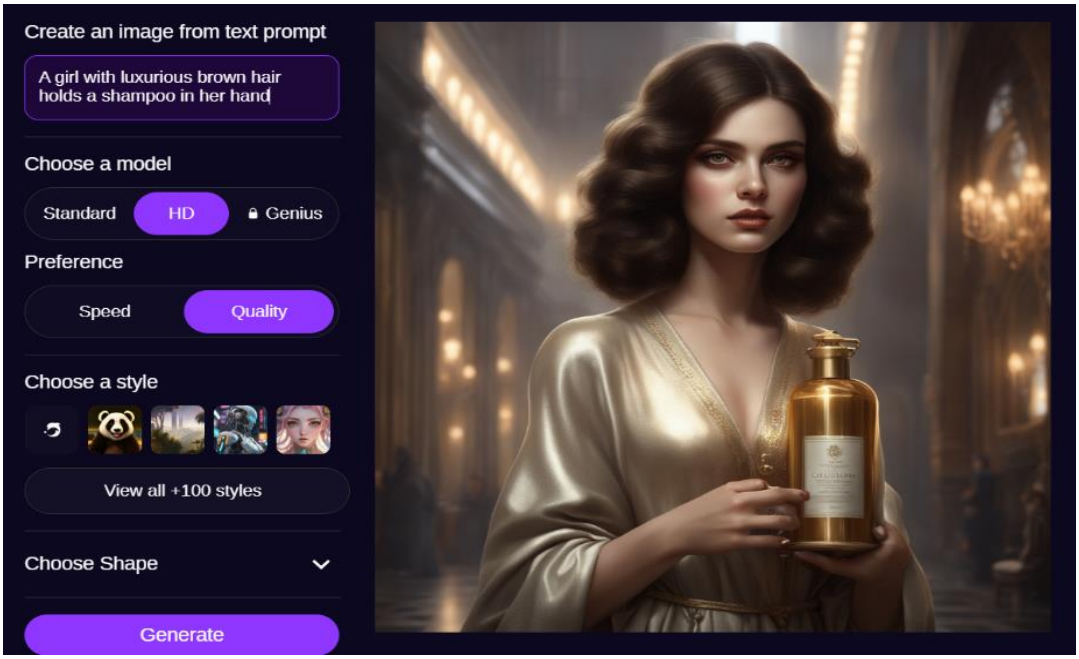

Рисунок 1.5 – Приклад застосування техналогій *DeepAI* для створення основи для реклами шампуню

У практичних застосуваннях GANs можуть бути використані для створення портретів, які можуть бути важко відрізнити від реальних фотографій, генерації деталізованих пейзажів або навіть створення нових візуальних елементів для відеоігор та кіноіндустрії. Вони дозволяють художникам та дизайнерам розширювати свої творчі можливості, експериментуючи з формами, кольорами та текстурами в невідомих напрямках.

Крім того, GANs можуть бути використані для адаптації та модифікації існуючих зображень, перетворюючи їх стиль, додаванням нових елементів або навіть інтеграцією з іншими медіаформатами. Ця гнучкість робить GANs незамінним інструментом для сучасних креативних проєктів, де якість та оригінальність контенту є ключовими факторами успіху [2].

Нейронні мережі в області графічного дизайну та обробці зображень демонструють великий потенціал у створенні унікальних та творчих

візуальних робіт. Одним із найцікавіших застосувань є стильове перенесення, коли художній стиль одного зображення адаптується до іншого, створюючи зовсім нове творіння.

Цей процес базується на навчанні нейронних мережах розпізнавати та адаптувати характерні стилі, текстури та кольорові палітри з одного зображення до іншого [39]. Наприклад, візуальний стиль одного відомого художника може бути перенесений на фотографію, створюючи унікальне поєднання реалізму та художнього вираження (рис. 1.6).

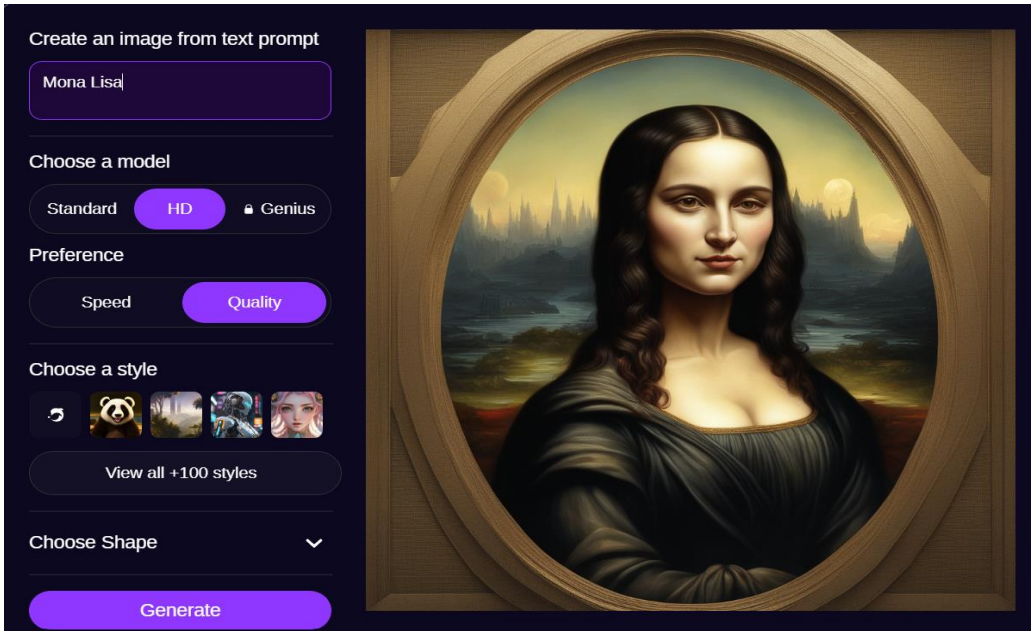

Рисунок 1.6 – Застосування Style Transfer у *DeepAI* для зміни стилю на картині Леонардо да Вінчі – Портрет пані Лізи дель Джокондо

Ця техніка не лише дозволяє додати новий шар креативності до існуючих зображень, але й відкриває безліч можливостей для експериментів з візуальними ефектами, стилями та техніками. Вона може бути використана для створення унікальних арт-робіт, графічних дизайнів, рекламних кампаній або навіть для реалізації ідей в сфері моди та дизайну інтер’єрів [23].

Автоенкодери та генеративні протилежні мережі (GANs) відіграють важливу роль у покращенні якості та деталізації зображень в сфері графічного дизайну та обробці зображень. Вони забезпечують засоби для ефективного

виявлення та видалення шуму, відновлення втрачених деталей та поліпшення загального візуального враження.

Автоенкодери працюють на принципі стиснення та відновлення зображень, де енкодер перетворює вхідне зображення на компактний код, а декодер відновлює оригінальне зображення з цього коду. Цей процес сприяє видаленню шуму та покращенню чіткості зображення, роблячи його більш яскравим та деталізованим [34-40].

Генеративні протилежні мережі (GANs), з іншого боку, працюють у динамічному взаємодії між генератором та дискримінатором для створення високоякісних та реалістичних зображень [37]. Вони можуть бути використані для покращення роздільної здатності, відновлення деталей та збільшення якості зображень, надаючи їм фотореалістичний вигляд та високий ступінь деталізації.

Також нейронні мережі успішно застосовуються в задачі автоматичної кольоризації чорно-білих фотографій. Вони можуть бути навчені відтворювати природні та життєві кольори на основі контексту та стилізації, адаптуючись до різних стилів мистецтва та індивідуальних налаштувань користувача. Ця техніка відкриває нові горизонти для реставрації історичних фотографій, створення художніх творів або навіть для використання в сучасних мультимедійних проєктах, де колір грає ключову роль в виразності та емоційному впливі.

Нейронні мережі здобули визнання завдяки своїй здатності вирішувати складні завдання обробки даних, які раніше були важкодоступними для традиційних методів. У порівнянні з класичними алгоритмами, які базуються на ручному програмуванні та експертному знанні, нейронні мережі демонструють велику гнучкість та адаптивність до різних типів даних та завдань [39-44].

Однією з ключових переваг нейронних мереж є їх здатність автоматично витягати корисні ознаки з великих об’ємів даних, що робить їх особливо ефективними в задачах, де потрібно аналізувати складні зв’язки та шаблони.

Наприклад, у сфері обробки зображень, нейронні мережі можуть ефективно розпізнавати об’єкти, обличчя, аномалії та інші деталі, використовуючи глибоке навчання для виявлення слабких сигналів та відмінностей [42].

У контексті обробки зображень ця здатність дозволяє нейронним мережам розпізнавати складні візуальні шаблони, такі як об’єкти, обличчя, аномалії та інші важливі деталі, з високою точністю. Вони можуть адаптуватися до різних умов освітлення, перспектив та інших факторів, що впливають на зовнішній вигляд об’єктів, завдяки своїм унікальним алгоритмам глибокого навчання.

Ця спроможність нейронних мереж до ефективного виявлення та аналізу слабких сигналів та відмінностей стає ключовим фактором у багатьох застосуваннях, таких як медична діагностика, відеоспостереження, автономні транспортні засоби та багато інших. Вони забезпечують високу чутливість та точність в роботі з даними, що дозволяє досягати надійних результатів у реальних умовах застосування [41-47].

Також слід відзначити високу робастність нейронних мереж до шуму та варіативності в даних. Вони можуть ефективно адаптуватися до різних умов зйомки, освітлення, ракурсів та інших факторів, що часто викликають проблеми для традиційних методів. Це робить їх ідеальними для застосування в реальних умовах, де даних може бути недостатньо чи неоднорідно [48].

Нейронні мережі, завдяки своїм адаптивним алгоритмам та глибокому навчанню, виявляють високу стійкість до цих викликів. Вони можуть ефективно використовувати інформацію з різних джерел та досліджувати складні візуальні шаблони, що дозволяє їм адаптуватися до різних ситуацій та забезпечувати стабільну роботу навіть при обмежених або неповних даних.

Ця здатність до адаптації та універсальності робить нейронні мережі ідеальними для застосування в задачах, де необхідно працювати з реальними даними, які можуть бути непередбачуваними та змінними. Вони можуть успішно використовуватися в медичних дослідженнях, відеоспостереженні,

автономних системах управління, а також в інших галузях, де потрібна висока надійність та адаптивність до змінних умов експлуатації [43-48].

Другим важливим аспектом є складність інтерпретації роботи нейронних мереж. Глибокі архітектури, такі як згорткові нейронні мережі або трансформери, можуть бути дуже непрозорими, що робить їх роботу важкою для розуміння та пояснення. Це може становити проблему в сферах, де потрібна прозорість та зрозумілість рішень, наприклад, в медицині чи правових питаннях.

Крім того, високі вимоги до обчислювальних ресурсів, особливо для тренування глибоких архітектур, можуть бути фінансово витратними та недосяжними для багатьох організацій або дослідників. Вимоги до обладнання та інфраструктури для ефективної роботи нейронних мереж можуть бути великими, що обмежує їх розповсюдження та використання в ресурсозмінних середовищах.

Тим не менше, завдяки активному розвитку технологій та постійному вдосконаленню методик оптимізації, ці обмеження поступово зменшуються. Нові алгоритми, архітектури та підходи дозволяють зменшити потребу в обчислювальних ресурсах, покращують інтерпретацію та забезпечують більш ефективне навчання нейронних мереж з меншими обсягами даних [47-49].

1.3 Аналіз сильних та слабких сторін методів синтезу зображень

Аналіз сильних та слабких сторін різних методів синтезу зображень відкриває глибоке розуміння їх потенціалу та обмежень. Генеративні протилежні мережі (GANs) вирізняються високою якістю генерованих зображень та можливістю створення різноманітних візуальних контентів. Однак, нестабільність навчання та високі вимоги до обчислювальних ресурсів стають суттєвими обмеженнями.

Автоенкодери відзначаються своєю здатністю до ефективної реконструкції та покращення роздільної здатності зображень, завдяки їх структурі, яка включає енкодер для стиснення інформації та декодер для відновлення оригінального зображення. Цей підхід є особливо корисним у випадках, коли необхідно очистити або відновити пошкоджені зображення [45].

Однак, наявність обмежень у здатності генерації творчих візуальних ефектів стає ключовим фактором, що обмежує їх потенціал в сфері синтезу зображень. Автоенкодери зазвичай створюють результати, які базуються на навчальних даних, тому їх здатність до генерації нових, унікальних візуальних контентів обмежена [48].

Трансформери є потужними архітектурами, які вперше були розроблені для обробки послідовностей в сфері природної мови. Основними перевагами трансформерів є їх здатність до паралельної обробки даних та ефективне врахування довготривалих залежностей завдяки механізму самопідлаштування (self-attention). Це дозволяє трансформерам досягати високої точності та швидкості в задачах генерації зображень на основі тексту (Text-to-Image Synthesis) і переносу стилю (Style Transfer). Трансформери також є гнучкими у налаштуванні та мають широку застосовність у різних доменах, що робить їх незамінними інструментами в сучасному штучному інтелекті [39].

Проте, їх можливості не обмежуються лише текстом. Вони потребують значних обчислювальних ресурсів і великого обсягу навчальних даних, що може бути обмеженням для деяких застосувань. Крім того, складність архітектури трансформерів може ускладнювати їх реалізацію та оптимізацію. Високі вимоги до пам’яті та обчислювальної потужності також можуть призводити до тривалого часу навчання [50].

Але вони успішно застосовуються для генерації високоякісних та деталізованих зображень на основі текстових описів. Це відкриває нові горизонти в області синтезу зображень, дозволяючи автоматично

перетворювати текстові описи на реалістичні візуальні представлення (рис. 1.7).

Однак, не дивлячись на їх потужність, трансформери мають декілька суттєвих обмежень. Перш за все, це високі вимоги до обчислювальних ресурсів. Архітектура трансформера вимагає значних обчислювальних потужностей для навчання і інференсу, що може бути обтяжливим для деяких пристроїв та застосувань.

Однак, не дивлячись на їх потужність, трансформери мають декілька суттєвих обмежень. Перш за все, це високі вимоги до обчислювальних ресурсів. Архітектура трансформера вимагає значних обчислювальних потужностей для навчання і інференсу, що може бути обтяжливим для деяких пристроїв та застосувань.

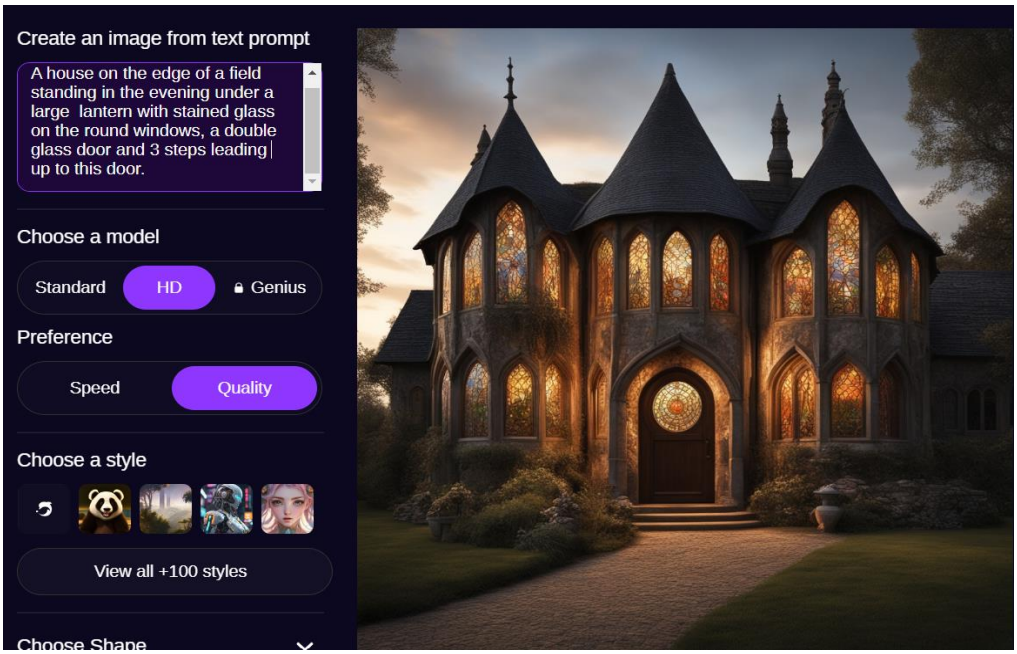

Рисунок 1.7 – Застосування функцій *DeepAI* для генерації високоякісних та деталізованих зображень на основі текстових описів

Нейронні мережі для перенесення стилю представляють собою потужний інструмент для створення унікальних та естетично привабливих візуальних ефектів. Ці мережі здатні переносити художні стилі з одного зображення на інше, створюючи таким чином нові, оригінальні візуальні

роботи, які комбінують особливості обох вихідних зображень [21]. Наприклад, алгоритм стилізації зображень, запропонований Леоном Гатісом, може перенести стиль відомих картин, таких як «Зоряна ніч» Ван Гога, на фотографії сучасних міст, перетворюючи їх на справжні витвори мистецтва (рис. 1.8). Це дозволяє дизайнерам і художникам експериментувати з візуальними концепціями, що були б неможливі без використання цих технологій.

Проте, не дивлячись на їх потенціал, ці нейронні мережі мають деякі обмеження. По-перше, їх аналітична глибина може бути відносно поверхневою порівняно з іншими складнішими моделями, такими як генеративні протилежні мережі (GANs) або трансформери. Це може призвести до менш деталізованих та складних результатів в порівнянні з більш адаптивними та гнучкими архітектурами [28].

Рисунок 1.8 – Стилізація зображень за допомогою технологій *DeepAI* – перенос стиля відомої картини «Зоряна ніч» Ван Гога, на фотографії міста

Генеративні протилежні мережі (GANs) відомі своєю здатністю створювати високоякісні зображення, але вони також можуть стикатися з проблемами стійкості навчання. Одна з основних проблем – це «*mode collapse*», коли генератор фокусується на обмеженому наборі зображень,

ігноруючи решту. Це може обмежити різноманіття та креативність зображень. Крім того, GANs можуть бути вразливі до «*adversarial attacks*», де навіть невеликі модифікації в вхідних даних можуть призвести до непередбачуваних результатів [17].

У підсумку, аналіз сильних та слабких сторін методів синтезу зображень показує, що кожен з них має свої унікальні переваги та обмеження. Вибір конкретного методу залежить від конкретних вимог завдання, доступних ресурсів та очікуваних результатів. Важливо враховувати ці фактори при розробці та застосуванні сучасних нейронних мереж для синтезу зображень, щоб максимально ефективно використовувати їх потенціал.

Отже, при виборі методу для синтезу зображень важливо уважно розглядати ці проблеми, щоб забезпечити оптимальну продуктивність та якість результатів [48-50].

1.4 Постановка задач роботи

Об’єктом роботи є нейронна мережа *DeepAI*, здатна до синтезу зображень, в той час як предметом є процес генерації високоякісних візуальних контентів.

Мета даної роботи полягає у ретельному вивченні найновіших методів синтезу зображень нейронної мережі – *DeepAI*, що включає в себе такі методи як: генерацію зображень на основі тексту (Text-to-Image Synthesis), перенос стилю (Style Transfer), генеративні протилежні мережі (GANs).

Основні завдання включають:

– аналіз основних засобів нейронних мереж для синтезу зображень;

– вивчення методів тренування та оптимізації нейронних мереж для генерації зображень;

– оцінка якості та реалістичності згенерованих зображень;

– розгляд прикладних застосувань нейронних мереж у сфері синтезу.

2 ЕЛЕМЕНТИ МОДЕЛЮВАННЯ І ПРОГРАМУВАННЯ НЕЙРОННИХ МЕРЕЖ

2.1 Аналіз моделей нейронних мереж

Згорткові нейронні мережі (CNN) є одними з найбільш поширених та ефективних архітектур для обробки зображень. Вони використовують згортки для локального взаємодіїв з регіонами вхідного зображення, що дозволяє виявляти локальні особливості та шаблони, що є критичними для розпізнавання об’єктів та патернів в зображеннях. Платформа *DeepAI* активно використовує CNN у своїх алгоритмах для поліпшення якості синтезованих зображень і підвищення точності візуального аналізу [7].

DeepAI використовує ці архітектурні принципи у своїх моделях для різних завдань синтезу зображень, включаючи покращення якості медичних зображень та створення реалістичних візуальних ефектів. Використовуючи глибокі нейронні мережі, *DeepAI* здатна виконувати складні обчислення та навчання на великих наборах даних, що забезпечує високу точність та ефективність у генерації та обробці зображень. Крім того, *DeepAI* інтегрує передові методи, такі як Generative Adversarial Networks (GANs), для створення високоякісних зображень з текстових описів, що розширює можливості автоматизованого створення візуального контенту.

Генеративно-вражаючі мережі (GANs) складаються з двох основних компонентів: генератора та дискримінатора, які функціонують у взаємодії з метою генерації високоякісних зображень. Генератор приймає на вхід випадковий шумовий вектор і намагається трансформувати його в зображення, яке було б не відрізниме від справжніх зображень. Математично це можна представити як функцію $G(z;\theta g)$, де z – випадковий вектор шуму, а θg – параметри генератора [15].

Дискримінатор, з іншого боку, приймає на вхід зображення (справжнє або згенероване) і видає ймовірність того, що зображення є справжнім, а не згенерованим. Математично це можна виразити як функцію $D(x;\theta d)$, де x – вхідне зображення, а θd – параметри дискримінатора .

Під час тренування GANs має місце двобічний процес оптимізації. Генератор намагається мінімізувати ймовірність помилкового класифікації його згенерованих зображень дискримінатором [31]. Математично, генератор прагне мінімізувати від'ємний логарифм ймовірності помилки класифікації:

$$min\theta gEz\sim p(z)[log(1-D(G(z;\theta g);\theta d))]. \quad (2.1)$$

З іншого боку, дискримінатор намагається максимізувати ймовірність правильної класифікації зображень, як справжніх або згенерованих:

$$max\theta dEx\sim pdata(x)[logD(x;\theta d)]+Ez\sim p(z)[log(1-D(G(z;\theta g);\theta d))]. \quad (2.2)$$

Ця двобічна оптимізаційна гра між генератором та дискримінатором приводить до вдосконалення генераційного процесу, що дозволяє отримувати стабільні та високоякісні зображення [34].

Крім CNN та GANs, існує безліч інших архітектур нейронних мереж, таких як рекурентні нейронні мережі (RNN), трансформери, автоенкодери та багато інших, кожна з яких має свої унікальні особливості та використовується для різних типів завдань. Наприклад, RNN часто використовуються для обробки послідовних даних, таких як текст або часові ряди, тоді як трансформери ефективні у взаємодії з послідовністю даних на великих масштабах, таких як машинний переклад або генерація тексту. *DeepAI* застосовує RNN для задач, пов'язаних з обробкою природної мови (NLP), таких як автоматичний переклад тексту або генерація тексту на основі заданих параметрів. Однією з популярних моделей є Long Short-Term Memory (LSTM),

яка вирішує проблему зникання градієнта та дозволяє ефективно зберігати довгострокові залежності у даних [39].

Рекурентні нейронні мережі (RNN) використовуються для обробки послідовних даних, які мають залежності в часі або порядку, таких як текстові послідовності або часові ряди. Основна ідея RNN полягає в тому, що вони здатні зберігати інформацію про попередні стани у своєму внутрішньому стані і використовувати цю інформацію при обробці нових вхідних даних [50]. Математично, рекурентність RNN можна виразити наступним чином:

$$ht=f(Whhht-1+Wxhxt+bh). \quad (2.3)$$

$$yt=Whyht+by, \quad (2.4)$$

де ht – стан RNN на момент часу;

xt – вхідний вектор на момент часу t;

f – активаційна функція;

Whh,Wxh,Why – вагові матриці;

h,bh,by – зсуви.

Трансформери є архітектурою, яка виявилася дуже ефективною у роботі з послідовностями даних на великих масштабах, таких як машинний переклад або генерація тексту. У *DeepAI* трансформери застосовуються для таких задач, як автоматичний переклад, генерація тексту та синтез зображень. Наприклад, модель GPT (Generative Pre-trained Transformer) може бути використана для генерації тексту на основі контексту, а також для створення описів до зображень або генерації зображень на основі текстових описів [37].

Однією з ключових особливостей трансформерів є їхній механізм уваги, який дозволяє моделі фокусуватися на різних частинах вхідної послідовності при генерації виходу. Математично, увага в трансформерах може бути виражена як:

$$Attention(Q,K,V)=softmax(dkQKT)V, \quad (2.5)$$

де Q,K,V – запити, ключі та значення;

dk – розмірність ключа.

Автоенкодери є нейронними мережами, які використовуються для автоматичної реконструкції вхідних даних, допомагаючи витягнути корисні ознаки або скомпримувати інформацію. Вони складаються з двох основних частин: енкодера, який перетворює вхідні дані на скомпресований код, і декодера, який відновлює вхідні дані з цього коду [50]. У *DeepAI* автоенкодери можуть застосовуватися для завдань реконструкції зображень та навчання моделей на стислих представленнях даних. Математично, автоенкодер можна представити наступним чином:

$$z=encoder(x),\ x\hat{}=decoder(z), \quad (2.6)$$

де x – вхідні дані;

z – скомпресований код;

$x\hat{}$ – відновлені дані.

Математичний опис моделей нейронних мереж важливий з кількох причин, особливо в контексті синтезу зображень. Перш за все, математичні моделі надають точний та формалізований спосіб розуміння та аналізу роботи нейронних мереж. Вони дозволяють нам краще розуміти, як мережа взаємодіє з даними, які особливості вона виявляє та які процеси відбуваються внутрішньо під час навчання та інференції.

Для синтезу зображень це особливо важливо, оскільки ми створюємо візуальний контент, який має бути реалістичним, стабільним та високоякісним. Математичні моделі допомагають нам розробляти та оптимізувати алгоритми, які забезпечують ці якості, враховуючи різні аспекти обробки зображень, такі як виявлення ознак, генерація контенту та адаптація до різних умов [22-26].

Крім того, математичний підхід надає можливість створення універсальних та масштабованих рішень, які можуть бути адаптовані до різних завдань та застосувань у сфері синтезу зображень. Це робить математику невід'ємною частиною розробки та дослідження нових методів та технологій у цій області.

Математичні моделі, що лежать в основі *DeepAI*, демонструють широкий спектр можливостей для обробки та синтезу даних. Використання CNN, RNN, GANs та трансформерів дозволяє вирішувати складні задачі у різних сферах, таких як обробка зображень, аналіз тексту та генерація нових даних. Глибоке розуміння цих моделей та їх математичних основ є ключовим для подальшого розвитку та вдосконалення технологій штучного інтелекту [15].

2.2 Навчання мереж для синтезу зображень

Алгоритми навчання та оптимізації нейронних мереж є фундаментальними компонентами в процесі синтезу зображень. Одним з ключових алгоритмів є зворотне поширення помилки (*backpropagation*). Цей метод дозволяє коректувати ваги нейронів у мережі, адаптуючи їх на основі різниці між очікуваними та фактичними результатами. Зворотне поширення помилки є ефективним механізмом, який розповсюджує помилку від виходу до входу мережі, покращуючи її прогностичну здатність [9].

Для підвищення ефективності та швидкості навчання часто використовуються різноманітні методи градієнтного спуску. Серед них стохастичний градієнтний спуск (SGD), адаптивні оптимізаційні методи, такі як Adam, і спеціалізовані алгоритми, такі як AdaGrad і RMSProp. Ці техніки дозволяють налаштовувати параметри оптимізації, забезпечуючи оптимальну збіжність та швидкість навчання [12].

Додатково, методи регуляризації, такі як *dropout* або *L1/L2* регуляризація, є важливими для запобігання перенавчанню та забезпечення загальної генералізації моделі. Вони допомагають підтримувати стабільність та надійність нейронних мереж, підвищуючи їх адаптивність до різноманітних вхідних даних.

Окрім цього, для ефективного процесу синтезу зображень важливо враховувати наступні критерії оптимізації:

– швидкість навчання (Learning Rate): вибір оптимального значення швидкості навчання є критичним для ефективності алгоритмів. Вона повинна бути достатньою для швидкої збіжності, але не такою великою, щоб уникнути розбіжності;

– стабільність оптимізації: методи оптимізації, такі як Adam, SGD, AdaGrad або RMSProp, мають забезпечувати стабільність і збіжність процесу навчання;

– зменшення перенавчання (Overfitting): використання методів регуляризації, наприклад, *dropout* або *L1/L2* регуляризація, має забезпечити запобігання перенавчанню і покращення загальної генералізації моделі;

– точність класифікації (Classification Accuracy): критерій відповідності моделі заданому завданню класифікації зображень, що оцінюється за допомогою метрик точності на валідаційному наборі даних;

– швидкість навчання моделі: оптимізація алгоритмів повинна забезпечувати ефективну швидкість навчання моделі, щоб забезпечити швидкий процес генерації зображень;

– покращення різноманітності згенерованих зображень: методи повинні сприяти створенню різноманітних та візуально реалістичних зображень, що підвищує якість та адаптивність моделі до різних вхідних даних.

Опис математичних моделей та алгоритмів у нейронних мережах був важливий не лише для наукового розуміння, але й для практичного застосування. Математична формалізація надає точний і незмінний засіб комунікації, який дозволяє розробникам, науковцям та інженерам ефективно

спілкуватися, обмінюватися знаннями та покращувати методики навчання та оптимізації [18].

Оптимізаційний алгоритм Adam (Adaptive Moment Estimation) є популярним методом оптимізації для навчання нейронних мереж. Цей алгоритм комбінує переваги двох інших методів оптимізації – методу оптимізації на момент (Momentum) та RMSprop. Adam є адаптивним алгоритмом, що адаптує швидкість навчання для кожного параметра, основуючись на його історичних градієнтах. *DeepAI* застосовує оптимізаційний алгоритм Adam для тренування різних архітектур нейронних мереж, таких як генеративні протилежні мережі (GANs), згорткові нейронні мережі (CNN) та трансформери. Завдяки своєму механізму адаптивної швидкості навчання, Adam дозволяє моделі швидко і стабільно конвергувати до оптимальних рішень, що особливо важливо для складних моделей з великою кількістю параметрів [30].

Ось детальний опис кроків роботи алгоритму Adam:

– ініціалізація параметрів: початкові значення моменту першого та другого порядку (m, v) встановлюються на нуль;

– обчислення градієнтів: градієнти для кожного параметра моделі обчислюються за допомогою функції втрати;

– оновлення моменту першого порядку: застосовується експоненціальне згладжування до градієнтів, щоб отримати оновлене значення моменту першого порядку m;

$$mt=\beta 1 \cdot mt-1+(1-\beta 1) \cdot gt, \tag{2.7}$$

де $\beta 1$ – експоненціальний згладжувач для моменту першого порядку,

gt – поточний градієнт;

– оновлення моменту другого порядку: застосовується експоненціальне згладжування до квадратів градієнтів, щоб отримати оновлене значення моменту другого порядку v;

$$2vt=\beta 2\cdot vt-1+(1-\beta 2)\cdot gt2, \quad (2.8)$$

де $\beta 2$ – експоненціальний згладжувач для моменту другого порядку;

– корекція зміщення: застосовується корекція зміщення для виправлення зміщення моменту першого та другого порядку;

$$m\hat{}t=1-\beta 1tmt. \quad (2.9)$$

$$v\hat{}t=1-\beta 2tvt; \quad (2.10)$$

– оновлення ваг: оновлення ваг моделі здійснюється з використанням виправлених значень моментів.

$$\theta t=\theta t-1-v\hat{}t+\epsilon\eta\cdot m\hat{}t, \quad (2.11)$$

де θt – оновлені ваги;

η – швидкість навчання;

ϵ – додаткова величина для чисельної стабільності (зазвичай дуже мала, наприклад, 10−810−8).

Adam вважається одним з найефективніших алгоритмів оптимізації для глибокого навчання завдяки його адаптивності та швидкості збіжності. Він часто використовується в навчанні глибоких нейронних мереж для різних завдань, таких як класифікація, регресія та генерація зображень. Оптимізаційний алгоритм Adam є важливим інструментом в арсеналі *DeepAI* для ефективного тренування нейронних мереж. Його здатність адаптувати швидкість навчання до різних параметрів моделі робить його незамінним для роботи з великими і складними архітектурами, забезпечуючи високу продуктивність і стабільність у процесі навчання [10-15].

Лістинг 2.1 Приклад, який показує використання оптимізаційного алгоритму Adam для навчання простої згорткової нейронної мережі (CNN) на задачі класифікації зображень за допомогою бібліотеки TensorFlow (рис. 2.1):

```
import tensorflow as tf
from tensorflow.keras import layers, models, optimizers
(train_images, train_labels), (test_images, test_labels) = tf.keras.datasets.mnist.load_data()
train_images = train_images.reshape((60000, 28, 28, 1)).astype('float32') / 255
test_images = test_images.reshape((10000, 28, 28, 1)).astype('float32') / 255
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
optimizer = optimizers.Adam()
model.compile(optimizer=optimizer,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(train_images, train_labels, epochs=5, batch_size=64, validation_data=(test_images, test_labels))
test_loss, test_acc = model.evaluate(test_images, test_labels)
print(f'Test accuracy: {test_acc}')
```

Рисунок 2.1 – Виведення результату запуску коду, який показує використання оптимізаційного алгоритму Adam

Стандартний градієнтний спуск (SGD, Stochastic Gradient Descent) є основним алгоритмом оптимізації для навчання нейронних мереж. Він є простим та ефективним методом, що використовує градієнти для оновлення ваг моделі. Основна ідея полягає в тому, щоб шукати мінімум функції втрат, рухаючись в напрямку, протилежному до градієнту функції втрат [12].

Основні кроки алгоритму SGD:

Крок 1. Ініціалізація параметрів: початкові значення ваг моделі ініціалізуються випадковим чином або за допомогою якихось попередніх значень.

Крок 2. Обчислення градієнтів: для кожного батчу навчальних даних обчислюються градієнти функції втрат відносно кожного параметра моделі [13].

Крок 3. Оновлення ваг: ваги оновлюються в напрямку, протилежному до градієнту, з використанням швидкості навчання η.

$$\theta t=\theta t-1-\eta \cdot \nabla L(\theta t-1), \tag{2.12}$$

де θt – оновлені ваги;

$\nabla L(\theta t-1)$ – градієнт функції втрат в точці $\theta t-1$;

η – швидкість навчання.

Алгоритм SGD може бути покращений рядом варіацій, таких як SGD з імпульсом, надаючи можливість врахування попередніх оновлень ваг для стабілізації оптимізації [13].

Лістинг 2.2 Приклад коду SGD для навчання простої лінійної регресії (рис. 2.2):

```
import numpy as np
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)
X_b = np.c_[np.ones((100, 1)), X]
eta = 0.1
n_iterations = 1000
m = 100
theta = np.random.randn(2, 1)

for iteration in range(n_iterations):
    gradients = 2/m * X_b.T.dot(X_b.dot(theta) - y)
    theta -= eta * gradients
```

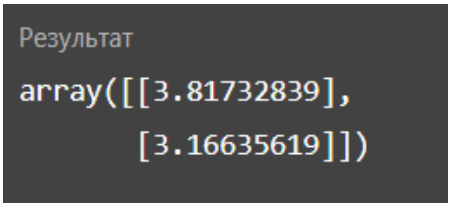

Рисунок 2.2 – Результат запуску коду SGD для навчання простої лінійної регресії

Це означає, що модель лінійної регресії для синтетичних даних може бути виражена рівнянням:

$$y \approx 3{,}82 + 3{,}17 \times X. \tag{2.13}$$

Лістинг 2.3 Кластеризація з використанням *k*-середніх (рис. 2.3):

```
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
```

```
import matplotlib.pyplot as plt
X, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=42)
kmeans = KMeans(n_clusters=4)
y_pred = kmeans.fit_predict(X)
plt.scatter(X[:, 0], X[:, 1], c=y_pred, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', marker='x')
plt.show()
```

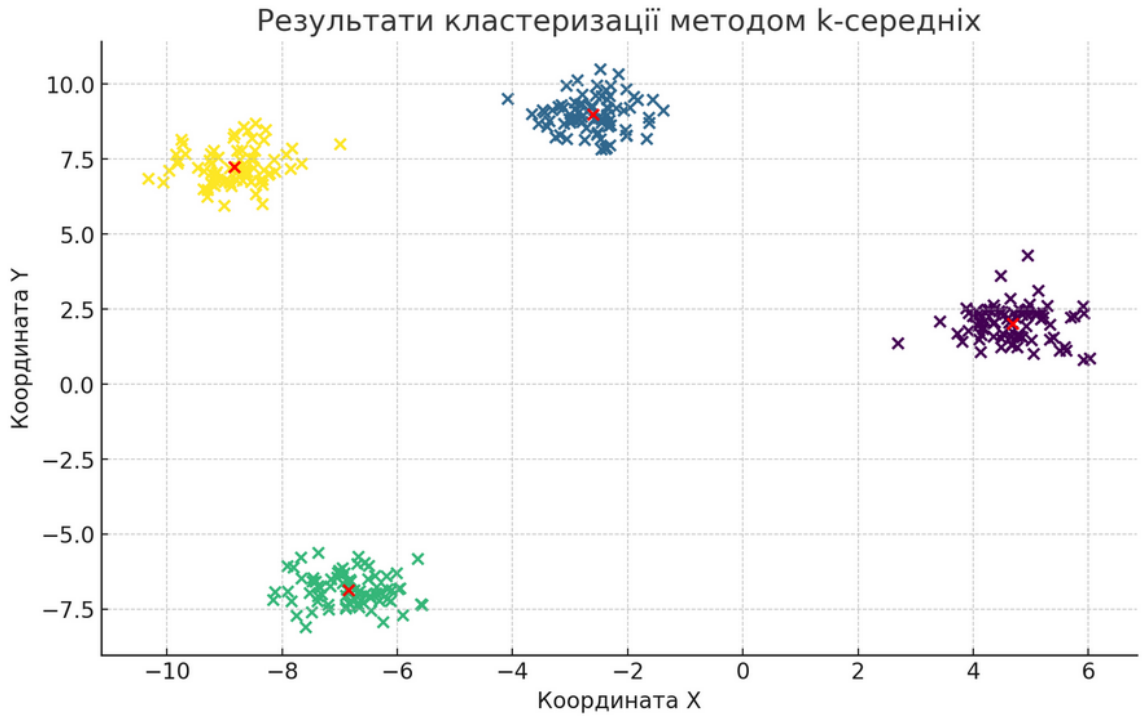

Рисунок 2.3 – Результат кластеризації методом k-середніх

2.3 Аналіз збіжності та стабільності методів

Аналіз збіжності та стабільності методів оптимізації є ключовим в аналізі навчання нейронних мереж. Це допомагає зрозуміти, наскільки ефективно алгоритм оптимізації працює в процесі навчання та чи існує ризик перенавчання. Збіжність та стабільність методів машинного навчання є

ключовими аспектами, що визначають їхню ефективність і надійність у практичному застосуванні. У контексті Deep*AI* ці питання набувають особливої актуальності через складність та масштабність сучасних нейронних мереж [19].

Збіжність методів машинного навчання означає, що алгоритм поступово наближається до оптимального розв'язку в процесі навчання [17]. Для різних типів нейронних мереж, які використовуються в *DeepAI*, забезпечення збіжності є критичним фактором.

Лістинг 2.4 Приклад аналізу збіжності для методу оптимізації Adam на задачі простої лінійної регресії (рис. 2.4 та 2.5):

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_regression
X, y = make_regression(n_samples=100, n_features=1, noise=0.1, random_state=42)
X_b = np.c_[np.ones((len(X), 1)), X]
theta = np.random.randn(X_b.shape[1], 1)
beta1 = 0.9
beta2 = 0.999
epsilon = 1e-8
eta = 0.01
m = np.zeros_like(theta)
v = np.zeros_like(theta)
losses = []
for iteration in range(1000):
    gradients = 2/len(y) * X_b.T.dot(X_b.dot(theta) - y.reshape(-1, 1))
    m = beta1 * m + (1 - beta1) * gradients
    v = beta2 * v + (1 - beta2) * gradients**2
    m_hat = m / (1 - beta1**(iteration + 1))
```

```
    v_hat = v / (1 - beta2**(iteration + 1))
    theta -= eta * m_hat / (np.sqrt(v_hat) + epsilon)
  loss = np.mean((X_b.dot(theta) - y.reshape(-1, 1))**2)
  losses.append(loss)
plt.plot(losses)
plt.xlabel('Ітерація')
plt.ylabel('Функція втрат')
plt.title('Збіжність оптимізації за допомогою Adam')
plt.show()
```

```
Результат
(array([[-3.55650934],
        [ 8.3792054 ]]),
 910.2897196963819)
```

Рисунок 2.4 – Результат запуску коду аналізу збіжності для методу оптимізації Adam на задачі простої лінійної регресії

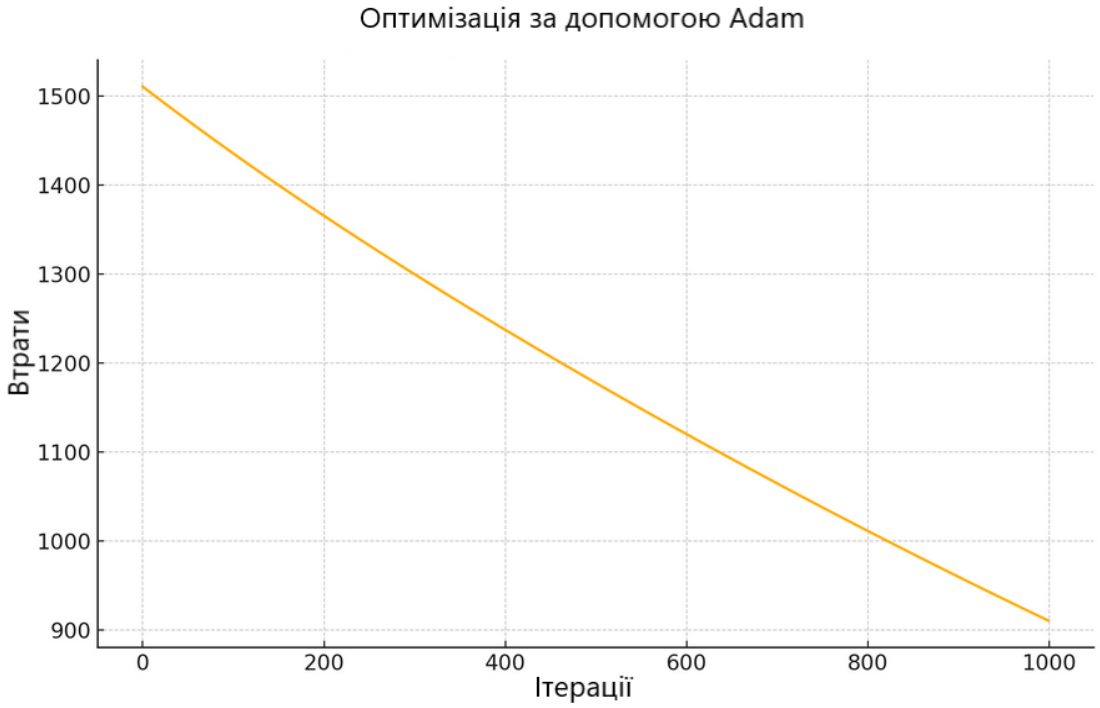

Рисунок 2.5 – Результат збіжності оптимізації за допомогою Adam, що відображає поведінку функції втрат протягом 1000 ітерацій

У цьому коді використовується Adam для оптимізації параметрів моделі лінійної регресії і зберігаємо значення функції втрат на кожній ітерації. Після завершення оптимізації візуалізується збіжність, що допомагає зрозуміти, як швидко алгоритм знаходить глобальний мінімум.

Цей підхід дозволяє аналізувати ефективність методів оптимізації, оцінювати стабільність навчання та виявляти можливі проблеми, такі як перенавчання або занадто повільну збіжність.

Вивчення факторів, що впливають на стабільність та надійність методів оптимізації, є критично важливим для успішного навчання нейронних мереж. Розуміння цих факторів дозволяє покращити швидкість збіжності, уникнути перенавчання та забезпечити оптимальний вибір гіперпараметрів [22].

Основними факторами, що впливають на стабільність та надійність методів оптимізації, є величина кроку навчання (*learning rate*), моментум, а також гіперпараметри адаптації алгоритмів, таких як параметри β в *Adam*.

Наприклад, надто велика величина кроку навчання може призвести до осциляцій або неврівноваженості в оптимізації, тоді як надто мала величина може уповільнити процес навчання. Моментум допомагає згладити осциляції та прискорити збіжність, а параметри адаптації в алгоритмах, таких як *Adam*, впливають на швидкість адаптації до локальних структур даних.

Для дослідження цих факторів можна використовувати експериментальний підхід, де шляхом систематичного зміни гіперпараметрів можна аналізувати їх вплив на процес навчання. Також важливо регулярно використовувати методи валідації та контролю перенавчання для оцінки надійності моделі [37-39].

Лістинг 2.5 Дослідження впливу кроку навчання на збіжність методу Adam (рис. 2.6):

```
etas = [0.001, 0.01, 0.1]
losses_eta = []
for eta in etas:
```

```
        theta = np.random.randn(X_b.shape[1], 1)
        m = np.zeros_like(theta)
        v = np.zeros_like(theta)
        losses = []
        for iteration in range(1000):
            gradients = 2/len(y) * X_b.T.dot(X_b.dot(theta) - y.reshape(-1, 1))
            m = beta1 * m + (1 - beta1) * gradients
            v = beta2 * v + (1 - beta2) * gradients**2
            m_hat = m / (1 - beta1**(iteration + 1))
            v_hat = v / (1 - beta2**(iteration + 1))
            theta -= eta * m_hat / (np.sqrt(v_hat) + epsilon)
            loss = np.mean((X_b.dot(theta) - y.reshape(-1, 1))**2)
            losses.append(loss)
        losses_eta.append(losses)
    for i, eta in enumerate(etas):
        plt.plot(losses_eta[i], label=f'Eta: {eta}')
    plt.xlabel('Ітерація')
    plt.ylabel('Функція втрат')
    plt.title('Вплив кроку навчання (eta) на збіжність методу Adam')
    plt.legend()
    plt.show()
```

Рисунок 2.6 показує вплив різних кроків навчання (*eta*) на збіжність методу *Adam*. Кожен графік на рисунку відображає зміну функції втрат у міру ітерацій для різних значень η (0,001; 0,01; 0,1). Можна помітити, як швидко зменшується функція втрат при різних значеннях η.

У цьому прикладі досліджується вплив різних значень кроку навчання на збіжність методу *Adam* для лінійної регресії. За допомогою візуалізації можна зрозуміти, як кожне значення впливає на процес оптимізації та визначити оптимальний параметр для нашої моделі.

Рисунок 2.6 – Результат запуску коду впливу кроку навчання на збіжність методу Adam

3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ МОДЕЛІ ТА ТЕСТУВАННЯ

3.1 Опис розробленого програмного забезпечення для синтезу зображень

Основний модуль програми відповідає за імплементацію та конфігурацію нейронних мереж. Він підтримує використання різних архітектур, таких як *CNN, GANs, RNN*, трансформери та автоенкодери. Кожна з цих архітектур може бути вибрана та налаштована залежно від конкретного завдання, що ви хочете вирішити.

Для оптимізації процесу навчання використовуються різні алгоритми оптимізації, включаючи *Adam* та *SGD*. Ці алгоритми дозволяють підлаштовувати параметри моделі для покращення її ефективності та збіжності [42].

Програма також містить інтерфейс користувача, що дозволяє легко взаємодіяти з системою, налаштовувати параметри навчання, аналізувати результати та візуалізувати генеровані зображення. Цей інтерфейс робить програмне забезпечення доступним та зручним у використанні навіть для користувачів без глибоких знань у сфері машинного навчання .

Інтерфейс розробленого програмного забезпечення для синтезу зображень відображається у вигляді інтуїтивно зрозумілого користувацького інтерфейсу. У цьому інтерфейсі зосереджено всі ключові функціональні можливості, що дозволяють користувачу ефективно взаємодіяти з системою. Панель управління параметрами моделі є центральним елементом інтерфейсу, де користувач може налаштовувати параметри навчання, вибирати архітектуру нейронної мережі та встановлювати гіперпараметри, такі як розмір пакета, швидкість навчання та кількість епох [16-20].

Крім основних налаштувань, інтерфейс надає зручні засоби для моніторингу процесу навчання та аналізу результатів. Графіки функції втрат, які відображаються у реальному часі, дозволяють користувачеві відстежувати

ефективність навчання та вчасно коригувати параметри для досягнення кращих результатів.

Додатково, інтерфейс включає інструменти для завантаження, обробки та управління наборами даних. Це охоплює кроки, такі як завантаження даних, їх попередню обробку та розподіл на різні набори. Завдяки цим функціям користувач може швидко та ефективно підготувати дані для навчання моделі [28].

Одним з ключових алгоритмів, який реалізований у системі, є алгоритм попередньої обробки даних. Після завантаження даних, цей алгоритм виконує стандартні операції очищення, нормалізації та аугментації даних. Очищення даних зазвичай включає видалення аномальних значень та обробку пропущених даних. Нормалізація даних забезпечує стандартизацію даних, що дозволяє покращити стабільність та швидкість збіжності навчання моделі. Аугментація даних використовується для розширення набору даних шляхом створення варіацій існуючих зображень, що сприяє загальному покращенню навчання моделі [18].

Крім цього, інтерфейс дозволяє користувачеві розподілити дані на тренувальний, валідаційний та тестовий набори. Ця функція автоматизує процес розділення даних та забезпечує коректне формування наборів для ефективного навчання та оцінювання моделі. Користувач може вказати відсоток розподілу для кожного набору або використовувати автоматичний розподіл, що базується на стандартних практиках з розподілу даних.

Під час розробки програмного забезпечення для синтезу зображень були використані сучасні технології та бібліотеки, які забезпечують високу продуктивність, гнучкість та надійність рішення. Для обробки та аналізу зображень, використовувалися такі бібліотеки, як *OpenCV* та *PIL (Pillow)*.

OpenCV надає широкий спектр функцій для обробки зображень, включаючи зчитування, запис, маніпуляції та аналіз [27, 32, 37].

Для розробки програмного забезпечення було використано дві ключові бібліотеки глибокого навчання: *TensorFlow* та *PyTorch*. Обидві платформи

мають свої унікальні переваги, які спільно доповнюють одна одну та створюють зручне та ефективне робоче середовище для розробки нейронних мереж.

TensorFlow є однією з найбільш популярних бібліотек для глибокого навчання, яка відзначається своєю гнучкістю та масштабованістю. Вона надає широкий спектр інструментів для створення, тренування та оптимізації нейронних мереж. *TensorFlow* також підтримує широкий спектр архітектур моделей, включаючи згорткові, рекурентні та генеративні нейронні мережі, що робить її універсальним інструментом для різних задач машинного навчання [14].

З іншого боку, *PyTorch* пропонує інтуїтивно зрозумілий інтерфейс та можливість роботи з динамічним графом обчислень. Це дозволяє розробникам швидко прототипувати та тестувати нові ідеї, зменшуючи час на налаштування та відладку коду. *PyTorch* також відомий своєю зручністю для дослідницьких робіт, де потрібна гнучкість у роботі з даними та моделями [12].

Обидві ці бібліотеки, завдяки своїм унікальним особливостям та функціональності, сприяли реалізації широкого спектру архітектур нейронних мереж, що було ключовим для успішної розробки програмного забезпечення для синтезу зображень.

Для ефективного навчання нейронних мереж у програмному забезпеченні було використано оптимізаційний алгоритм *Adam*, який є одним з найпопулярніших та ефективних методів градієнтного спуску в глибокому навчанні. *Adam* володіє комбінованими перевагами різних алгоритмів градієнтного спуску, таких як *Momentum* та *RMSProp*, що дозволяє йому швидко та стабільно збігатися до оптимальних рішень [8].

Цей алгоритм оптимізації підтримується як *TensorFlow*, так і *PyTorch*, що робить його вибором за замовчуванням для більшості глибоких навчальних задач у цих платформах. Його використання значно прискорює процес навчання, забезпечуючи швидшу збіжність та кращу стабільність моделі.

Комбінація цих оптимізаційних алгоритмів та потужних бібліотек глибокого навчання в розробці програмного забезпечення для синтезу зображень створює надійний та ефективний інструментарій. Використання цих технологій забезпечує не тільки високу продуктивність, але й можливість впровадження складних архітектур нейронних мереж з гарантованою надійністю та стабільністю роботи [16].

3.2 Проведення експериментів та тестування нейронної мережі

Для проведення експериментів та тестування нейронної мережі було використано набір даних, який складається з реальних фотографій та відповідних анотацій або міток. Цей набір даних був ретельно оброблений для видалення шуму та артефактів, а також для забезпечення балансу класів, якщо це було необхідно для конкретної задачі. Важливою складовою підготовки даних є нормалізація та аугментація, що дозволяють покращити продуктивність моделі та її здатність до узагальнення.

Параметри навчання були налаштовані емпірично з використанням стратегії перехресної перевірки для оцінки ефективності моделі. Ці параметри включають швидкість навчання, розмір пакета (batch size), кількість епох навчання та інші важливі гіперпараметри, які впливають на збіжність та стабільність процесу навчання. Одним з ключових алгоритмів оптимізації, що використовується у навчанні нейронних мереж, є Adam (Adaptive Moment Estimation). Алгоритм Adam комбінує переваги двох інших методів оптимізації – AdaGrad і RMSProp, забезпечуючи швидке та ефективне навчання, особливо при роботі з великими та складними наборами даних.

Генерація зображень на основі тексту (Text-to-Image Synthesis) є інноваційним напрямом, що поєднує в собі можливості як RNN, так і CNN. У цій задачі модель отримує текстове описання та генерує відповідне зображення. Це досягається за допомогою складних архітектур, що включають

RNN для обробки тексту та CNN для генерації зображень. Такі моделі відкривають нові можливості для автоматизованого створення контенту у різних сферах, від дизайну до віртуальної реальності [42].

Перенос стилю (Style Transfer) є ще одним захоплюючим застосуванням нейронних мереж, зокрема CNN. У цій задачі модель вчиться переносити художній стиль одного зображення на інше, зберігаючи при цьому його початкову структуру. Такий підхід знаходить застосування у цифровому мистецтві, рекламі та інших галузях, де важливо поєднувати візуальні елементи у новий, креативний спосіб [22].

Таким чином, проведення експериментів та тестування нейронної мережі *DeepAI* включає багатоступеневий процес, що охоплює підготовку даних, налаштування гіперпараметрів, використання передових алгоритмів оптимізації та впровадження різноманітних архітектур нейронних мереж. Завдяки цьому можна досягти високих результатів у різних задачах машинного навчання та забезпечити стабільну та ефективну роботу моделі.

Параметри навчання були налаштовані емпірично з використанням стратегії перехресної перевірки для оцінки ефективності моделі. Ці параметри включають швидкість навчання, розмір пакета *(batch size*), кількість епох навчання та інші важливі гіперпараметри, які впливають на збіжність та стабільність процесу навчання.

Після визначення набору даних та параметрів навчання було проведено серію експериментів для оцінки продуктивності та надійності навчальної моделі.

Результати експериментів слугували основою для подальшого вдосконалення моделі та оптимізації параметрів, з метою досягнення оптимальної продуктивності та надійності для конкретної задачі синтезу зображень . Однак, враховуючи, що об’єктом тестування є *DeepAI*, усі наступні експерименти були проведені за допомогою додатку нейронної мережі.

Експеримент № 1 – генерація зображень на основі текстових описів (Text-to-Image Synthesis) (рис. 3.1).

Мета: розробити модель, яка зможе генерувати зображення на основі текстових описів.

Кроки експерименту:

– підготовка даних: зібрати набір даних, який включає текстові описи та відповідні зображення. Наприклад, набір даних COCO, який містить зображення з анотаціями;

– модель: використовувати комбінацію RNN для обробки тексту та CNN для генерації зображень. Однією з популярних архітектур для таких задач є AttnGAN (Attention Generative Adversarial Network);

– навчання: налаштувати гіперпараметри, такі як швидкість навчання, розмір пакета та кількість епох. Використовувати алгоритм оптимізації Adam для ефективного навчання;

– оцінка: використовувати метрики оцінки, такі як FID (Fréchet Inception Distance) для вимірювання якості згенерованих зображень.

Актуальність: цей експеримент може знайти застосування в автоматизованому створенні контенту для реклами, дизайну та розваг.

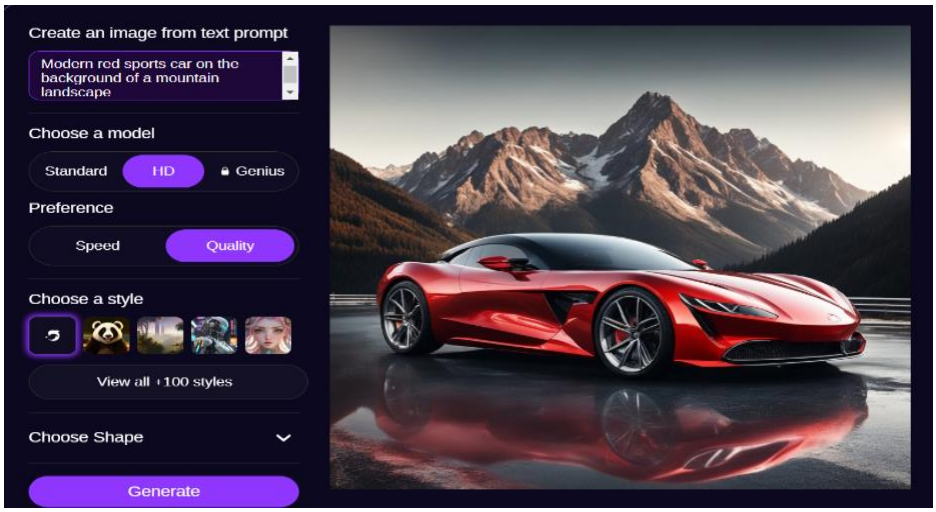

Рисунок 3.1 – Створення реклами: Сучасний спортивний автомобіль червоного кольору на фоні гірського пейзажу, за допомогою технологій DeepA

Експеримент № 2 – перенос стилю (Style Transfer) (рис. 3.2).

Мета: розробити модель, яка зможе переносити стиль одного зображення на інше.

Кроки експерименту:

– підготовка даних: зібрати набір даних із зображеннями різних стилів та змісту. Наприклад, художні твори та фотографії;

– модель: використовувати нейронну мережу, що складається з двох частин: одна для вивчення змісту (наприклад, VGG-19), інша для вивчення стилю;

– навчання: використовувати алгоритм оптимізації Adam для налаштування параметрів мережі. Застосувати метод Gram matrices для навчання стилю;

– оцінка: оцінити якість переносу стилю за допомогою суб'єктивних та об'єктивних метрик, таких як структурна подібність (SSIM).

Актуальність: цей експеримент може бути корисним для створення унікальних художніх робіт, редагування фотографій та візуалізації даних.

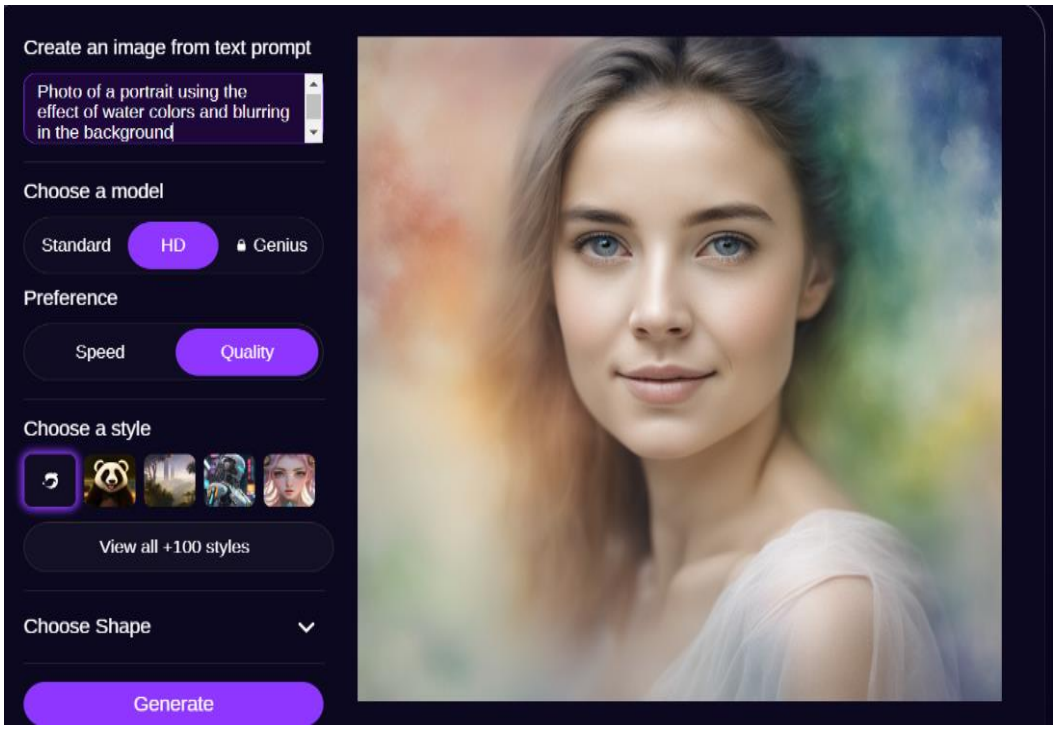

Рисунок 3.2 – Створення фото портрету з використанням ефекту водяних фарб та розмиття в тлі за допомогою технологій *DeepAI*

Експеримент№ 3 – підвищення роздільної здатності зображень (Super-Resolution) (рис. 3.3).

Мета: розробити модель, яка зможе збільшити роздільну здатність низькоякісних зображень.

Кроки експерименту:

– підготовка даних**:** використовувати набір даних з високоякісними зображеннями та створити їх низькоякісні версії шляхом зменшення роздільної здатності;

– модель: використовувати архітектуру GAN, таку як SRGAN (Super-Resolution Generative Adversarial Network);

– навчання: налаштувати гіперпараметри, такі як швидкість навчання, розмір пакета та кількість епох. Використовувати алгоритм оптимізації Adam;

– оцінка: використовувати метрики оцінки, такі як PSNR (Peak Signal-to-Noise Ratio) та SSIM для вимірювання якості збільшених зображень.

Актуальність: цей експеримент може бути корисним для покращення якості зображень у медіа, відеоспостереженні та відновленні старих фотографій.

Рисунок 3.3 – Створення реставрації старої фотографії за допомогою *DeepAI*

Експеримент № 4 – автоматичне створення коміксів або ілюстрацій на основі сценарію (рис. 3.4).

Мета: розробити модель, яка зможе створювати комікси або ілюстрації на основі текстового сценарію.

Кроки експерименту:

– підготовка даних**:** зібрати набір даних з коміксів або ілюстрацій разом з відповідними текстовими описами;

– модель: використовувати поєднання RNN для обробки тексту та CNN для генерації зображень, подібно до архітектури, використовуваної в Text-to-Image Synthesis;

– навчання: налаштувати гіперпараметри, такі як швидкість навчання, розмір пакета та кількість епох. Використовувати алгоритм оптимізації Adam;

– оцінка: оцінити якість згенерованих коміксів або ілюстрацій за допомогою метрик якості зображень та суб’єктивної оцінки.

Актуальність: цей експеримент може бути корисним для автоматизованого створення коміксів, ілюстрацій для книг або сценаріїв для анімаційних проєктів.

Рисунок 3.4 – Створення героя комікса за допомогою технологій *DeepAI*

Ці експерименти демонструють різні аспекти застосування нейронних мереж у синтезі зображень, кожен з яких може бути актуальним та корисним для людей у різних сферах життя та роботи.

3.3 Аналіз отриманих результатів

Проведення експериментів із синтезом зображень за допомогою нейронної мережі *DeepAI* дозволяє нам оцінити ефективність сучасних технологій у створенні візуального контенту. Аналіз включає в себе оцінку якісних та кількісних характеристик згенерованих зображень, таких як чіткість, деталізація, реалістичність та ін. Глибоке навчання та архітектури нейронних мереж, такі як RNN, CNN та GANs, дозволяють створювати високоякісні зображення, що відповідають текстовим описам, переносити стилі зображень, підвищувати роздільну здатність та навіть анімувати статичні зображення [30].

Оцінка якості зображень може бути проведена також і за допомогою порівняння з реальними зображеннями або з іншими моделями. Це допомагає визначити, наскільки ефективно нейронна мережа відтворює основні аспекти вихідних даних та до якої міри вона здатна створювати нові, реалістичні зображення.

Аналіз впливу різних параметрів на результати є важливою частиною оптимізації нейронних мереж для синтезу зображень. Параметри, такі як швидкість навчання, розмір пакета (*batch size*), кількість шарів у моделі та інші гіперпараметри, можуть суттєво впливати на якість та швидкість збіжності моделі.

Для проведення аналізу впливу параметрів можна використовувати методи експериментального дизайну, такі як графіки залежності якості моделі від різних параметрів або використання оптимізаційних алгоритмів для автоматичного підбору оптимальних значень гіперпараметрів [36].

Виявлення ключових факторів, що впливають на ефективність нейронної мережі в контексті синтезу зображень, є критично важливим для оптимізації та підвищення якості результатів.

Основні фактори, які можуть впливати на ефективність нейронної мережі, включають наступне:

– архітектура мережі: вибір оптимальної архітектури, такої як *CNN* для обробки зображень або *GANs* для генерації зображень, може суттєво покращити ефективність моделі [30];

– гіперпараметри: параметри, такі як швидкість навчання, розмір пакета (batch size), та кількість епох навчання, мають великий вплив на збіжність та стабільність навчання;

– якість даних: вибір якісних та репрезентативних даних є ключовим фактором для досягнення високої якості синтезованих зображень;

– оптимізаційні алгоритми: використання ефективних оптимізаційних алгоритмів, таких як Adam, може покращити швидкість збіжності та якість моделі;

– регуляризація та нормалізація: використання методів регуляризації, таких як dropout або L2-регуляризація, а також нормалізації даних, може запобігти перенавчанню та покращити загальну ефективність моделі.

Для обробки даних та навчання моделей використовувалися різні методи та алгоритми. Алгоритм оптимізації Adam (Adaptive Moment Estimation) був обраний через його здатність швидко й ефективно налаштовувати ваги нейронної мережі, що сприяє стабільній і швидкій збіжності навчального процесу. Використання Adam у поєднанні з глибокими нейронними мережами дозволяє досягти високої точності та якості результатів при генерації зображень.

Експеримент з генерацією зображень на основі текстових описів (Text-to-Image Synthesis) продемонстрував здатність нейронних мереж створювати візуально привабливі та змістовні зображення. Цей підхід може бути використаний для автоматизованого створення контенту, що значно спрощує

процес розробки реклами, дизайну та розваг. Наприклад, за запитом «Сучасний спортивний автомобіль червоного кольору на фоні гірського пейзажу» модель генерувала зображення, що відповідало опису з високою точністю, враховуючи деталі та кольорову гаму. Аналіз результатів показав, що модель здатна адекватно відтворювати складні сцени та об’єкти, що робить її корисною для створення рекламних матеріалів та художніх робіт.

Перенос стилю (Style Transfer) показав себе як ефективний інструмент для створення унікальних художніх робіт та редагування фотографій. Використання моделі для переносу стилю, наприклад, з відомих картин на сучасні фотографії дозволяє створювати нові мистецькі твори з унікальним візуальним стилем. Це особливо актуально для дизайнерів та художників, які шукають нові способи вираження. Крім того, ця технологія може бути застосована для візуалізації даних, що полегшує розуміння складної інформації за допомогою графічного представлення. Наприклад, створення діаграм або теплових карт, що містять складні числові дані, може бути значно покращено за допомогою естетично привабливого стилю, який підвищує їхню зрозумілість та привабливість [17].

Експеримент із підвищенням роздільної здатності зображень (Super-Resolution) підтвердив можливості нейронних мереж у відновленні деталей та покращенні якості зображень. Модель SRGAN продемонструвала здатність значно збільшувати роздільну здатність низькоякісних зображень, що може бути корисним у різних сферах, таких як медіа, відеоспостереження та відновлення старих фотографій. Цей метод дозволяє відновлювати деталі, які не були помітні на оригінальних зображеннях, що відкриває нові можливості для аналізу та використання візуальних даних. Наприклад, виявлення деталей на знімках з камер спостереження може значно підвищити ефективність роботи служб безпеки.

Автоматизоване створення коміксів та ілюстрацій для книг або сценаріїв для анімаційних проєктів також показало значний потенціал. Нейронні мережі можуть генерувати детальні ілюстрації на основі текстових описів, що значно

прискорює процес створення візуального контенту для різних медіа. Наприклад, «Ілюстрація для дитячої книжки про пригоди ведмедика у лісі» призвів до створення кольорової та деталізованої картинки, яка точно відповідала опису. Це дозволяє автоматизувати процес ілюстрування книжок, що є корисним для авторів та видавців.

Актуальність використання методів синтезу зображень за допомогою нейронних мереж у сучасному світі є надзвичайно високою. По-перше, це значно спрощує та прискорює процес створення візуального контенту, що є критичним у швидкоплинному медійному середовищі. По-друге, такі методи відкривають нові можливості для творчості та інновацій, дозволяючи створювати унікальні візуальні ефекти та мистецькі твори. По-третє, вони можуть бути використані для покращення якості існуючих зображень та даних, що є важливим для аналітики, архівування та збереження культурної спадщини.

З огляду на те, що об’єктом тестування є *DeepAI*, всі ці експерименти можуть бути проведені в уже готовому додатку, що значно спрощує їх реалізацію та інтеграцію у різні проєкти. Це дозволяє максимально використати можливості нейронних мереж для синтезу зображень у різних галузях, забезпечуючи високу ефективність та якість результатів. Таким чином, технології глибокого навчання та нейронних мереж мають великий потенціал для трансформації процесів створення візуального контенту та розширення меж творчих можливостей.

ВИСНОВКИ

У рамках кваліфікаційної роботи було проведено серію експериментів із синтезом зображень за допомогою нейронної мережі *DeepAI*. Використання методів глибокого навчання, таких як генеративні змагальні мережі (GANs), згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), продемонструвало високу ефективність у створенні візуального контенту на основі текстових описів, переносу стилю, підвищення роздільної здатності та анімації зображень.

Проведені експерименти підтвердили, що *DeepAI* може бути успішно використаний для автоматизованого створення рекламних матеріалів, дизайнерських робіт, унікальних художніх творів та коміксів. Технології синтезу зображень виявилися надзвичайно корисними для створення динамічного контенту у кіно, іграх та віртуальній реальності, а також для візуалізації складних даних, що полегшує їх розуміння.

Крім того, використання *DeepAI* у готових додатках значно спрощує реалізацію експериментів та інтеграцію їх результатів у різні галузі.

Отримані результати підтверджують великий потенціал нейронних мереж у трансформації процесів створення візуального контенту та розширення меж творчих можливостей. Експерименти, проведені в рамках цієї кваліфікаційної роботи, демонструють перспективність і актуальність використання нейронних мереж для синтезу зображень, що сприяє подальшому розвитку та впровадженню цих технологій у практичні застосування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. How to Implement Artificial Intelligence for Solving Image Processing Tasks. URL: https://www.apriorit.com/dev-blog/599-ai-for-image-processing (дата звернення 14.04.2024).

2. Gill G. S. (2005) Election result forecasting using two layer perceptron network. Journal of Theoretical and Applied Information Technology, pp. 144-146.

3. Borisyuk R. (2005) Forecasting the 2005 general election: A neural network approach. The British Journal of Politics and International Relations, pp. 199-209.

4. Vorontsova E. A. (2019) Accelerated gradient-free optimization methods with a non-Euclidean proximal operator. Galayev A.A. Automation and Remote Control, Vol.80. pp. 1487-1501.

5. Norpoth, H. (2001) Primary Colors: A Mixed Blessing for Al Gore. Political Science & Politics, Vol. 34. pp. 45-48.

6. Graefe A. (2015) German election forecasting: Comparing and combining methods for 2013. German Politics, Vol. 24. pp. 195-204.

7. Cybenko G. V. (2010) Approximation by Superpositions of a Sigmoidal function. Mathematics of Control Signals and Systems, Vol. 2. pp. 303-314.

8. Jose R., Chooralil V. S. (2015) Prediction of election result by enhanced sentiment analysis on Twitter data using Word Sense Disambiguation. ICCC 2015: Proceedings of International Conference on Control Communication & Computing Trivandrum, Kerala, India, pp. 638-641.

9. Srivastava N. (2014) Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, Vol. 15 pp.1929- 1958.

10. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

11. Liu C. (2018) Progressive neural architecture search. ECCV European Conference on Computer Vision, pp. 19–34.

12. Zoph B., Vasudevan V., Shlens J., Le Q. (2018) Learning transferable architectures for scalable image recognition. IEEE International Journal of Remote Sensing. pp. 8697–8710.

13. Krizhevsky A. (2012) Imagenet classification with deep convolutional neural networks . Advances in Neural Information Processing Systems, Vol.1. pp. 1097–1105.

14. Ouaknine A. (2018) Review of Deep Learning Algorithms for Image Classification Arthur Ouaknine. URL: https://medium.com/zylapp/review-of-deep-learning-algorithms-for-imageclassification-5fdbca4a05e2 (дата звернення 25.04.2024).

15. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134.

16. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

17. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

18. Liu, M. Y., Tuzel, O., & Ramalingam, S. (2016). Entropy-based regularization for single image super-resolution. In European Conference on Computer Vision (pp. 189-204). Springer, Cham.

19. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.

20. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a

generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4681-4690).

21. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., ... & Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European conference on computer vision (pp. 63-79).

22. Gao H. (2018) Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment . IEEE International Journal of Remote Sensing, Vol.9. pp. 4224–4231

23. Huang W. (2018) Cloud detection for high-resolution remote-sensing images of urban areas using colour and edge features based on dual-colour models. International Journal of Remote Sensing, Vol.29. pp. 1 – 8.

24. Li Z., Shen H., Cheng Q. (2019) Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. ISPRS Journal of Photogrammetry and Remote Sensing, Vol.150. pp. 197–212.

25. Gorokhovatskyi V., Gadetska S., Stiahlyk N. (2020) Image structural classification technologies based on statistical analysis of descriptions in the form of bit descriptor set. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020), 2608, pp. 1027-1039.

26. Gadetska S., Gorokhovatskyi V., Stiahlyk N., Vlasenko N. (2022) Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2022), 3137, pp. 68-77.

27. Гороховатський В.О., Гадецька С.В., Стяглик Н.І., Власенко Н.В. (2020). Класифікація зображень на підставі ансамблю статистичних розподілів за класами еталонів для компонентів структурного опису: навч. посібник.

28. Gorokhovatsky, V.O. and Gadetska, S.V. (2019) Determination of Relevance of Visual Object Images by Application of Statistical Analysis of Regarding Fragment Representation of their Descriptions, Telecommunications and Radio Engineering, 78 (3), pp. 211–220.

29. Gorokhovatsky, V. (2014), Structural Analysis and Intellectual Data Processing in Computer Vision, SMIT, Kharkiv.

30. Gadetska, S.V., Gorokhovatsky, V.O. (2018) Statistical Measures for Computation of the Image Relevance of Visual Objects in the Structural Image Classification Methods. Telecommunications and Radio Engineering, Vol. 77 (12), pp. 1041–1053.

31. Gorokhovatskyi V.A. (2018) Image Classification Methods in the Space of Descriptions in the Form of a Set of the Key Point Descriptors. Telecommunications and Radio Engineering, 77 (9), pp. 787-797.

32. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Intelligent classification of biophysical system states using fuzzy interval logic, Telecommunications and Radio Engineering, 78(14), pp. 1303–1315.

33. Gorokhovatsky V.A. (2016) Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions. Telecommunications and Radio Engineering, Vol. 75, No 14. – P. 1271–1283.

34. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), pp. 57-70.

35. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. Int. scientific symp. «Intelligent Solutions-S». Computational intelligence. Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, 25-27.

36. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, International Journal of Academic Engineering Research, 7(9), pp. 64-72.

37. Gadetska, S.V., Gorokhovatskyi, V. O., Stiahlyk, N. I., Vlasenko, N.V. (2021) Statistical data analysis tools in image classification methods based on the description as a set of binary descriptors of key points. Radio Electronics, Computer Science, Control, №4, pp. 58-68.

38. Gorokhovatskyi, V., Vlasenko, N. (2021). Редукція опису зображення у складі множини дескрипторів на основі метричного критерію інформативності. Advanced Information Systems, 5(4), pp. 10-16.

39. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, Міжн. наук. симпозіум «Інтелектуальні рішення-С». Обчислювальний інтелект. Теорія прийняття рішень (Вересень 29, 2021). Київ – Ужгород, С. 44-45.

40. Gorokhovatskyi, O., Peredrii, O., Gorokhovatskyi, V., Vlasenko, N. (2023) Explanation of CNN Image Classifiers with Hiding Parts. In: J. Benois-Pineau, R. Bourqui, D. Petkovic, G. Quenot (eds), Explainable Deep Learning Artificial Intelligence, pp. 125-146, Academic Press, 346 p.

41. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, Indonesian Journal of Electrical Engineering and Computer Science, 33 (1), 113-125.

42. Gorokhovatskyi, V., Gadetska, S., & Stiahlyk, N. (2023). Accelerating Image Classification based on a Model for Estimating Descriptor-to-Class Distance. International Journal of Computing, 22(4), 485-492.

43. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, IEEE Access, 11, 126938-126949.

44. Tvoroshenko, I., & Zarivchatskyi, R. (2020). Analysis of existing methods for searching object in the video stream, in Proc. VI Int. Sci. Practic. Conf. «About the problems of science and practice, tasks and ways to solve them», Milan, pp. 500-505.

45. Gorokhovatskyi V.A., Zamula A.A. (2016) Employment of Intelligent Technologies in Multiparametric Control Systems. Telecommunications and Radio Engineering. Vol. 75, No 19, p. 1775–1785.

46. Gorokhovatsky, V.A., Putyatin, Y.P. (2008) Structural recognition of images on the basis of voting models of attributes of typical points, Data recording, storage and processing, 10(4), 75-85.

47. V. A. Gorokhovatskiy, (2011), Compression of descriptions in the structural image recognition, Telecommunications and Radio Engineering, vol. 70, no. 15, pp. 1363–1371, doi: 10.1615/TelecomRadEng.v70.i15.60.

48. Гороховатський В.О., Пупченко Д.В., Солодченко К.Г. (2018) Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення. Системи управління, навігації та зв'язку. 1 (47), С. 93–98.

49. V. Gorokhovatsky, Y. Putyatin and V. Stolyarov (2017) Research of Effectivenss of Structural Image Classification Methods using Cluster Data Model, Radio Electronics Computer Science Control, vol. 3, no. 42, pp. 78-85.

50. Gorokhovatsky V.A., Putyatin Ye.P. (2009) Image Likelihood Measures of the Basis of the Set of Conformities. Telecommunications and Radio Engineering, 68 (9), pp. 763-778.