

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ МІЖНАРОДНИХ ВІДНОСИН
КАФЕДРА КОМП'ЮТЕРНИХ МУЛЬТИМЕДІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ О.А. Бобарчук
« ____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР

ЗА СПЕЦІАЛЬНІСТЮ 186 «ВИДАВНИЦТВО ТА ПОЛІГРАФІЯ»

Тема: «Засоби впровадження генеративних зображень за допомогою машинного навчання моделей (в контексті створення мультимедійного контенту)»

Виконавець _____ студент групи ВП-213М Лебедев Вадим Сергійович
(студент, група, прізвище, ім'я, по батькові)

Керівник _____ к.т.н. Бобарчук О.А.
(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер: _____ Гальченко С.М.
(підпис) (ПІБ)

КИЇВ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет міжнародних відносин

Кафедра комп'ютерних мультимедійних технологій

Спеціальність, спеціалізація 186 «Видавництво та поліграфія»

(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.А. Бобарчук

« » 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Лебедева Вадима Сергійовича

(П.І.Б. випускника)

1. Тема кваліфікаційної роботи: «Засоби впровадження генеративних зображень за допомогою машинного навчання моделей (в контексті створення мультимедійного контенту)» затверджена наказом ректора від 22.09.2023 №1901/ст.
2. Термін виконання роботи: з 02.10.2023 р. по 31.12.2023 р.
3. Вихідні дані до роботи: ілюстративний матеріал генеративних зображень на базисі моделей машинного навчання, верстка відповідного веб-ресурсу.
4. Зміст пояснювальної записки: теоретичні засади архітектури та впровадження моделей машинного навчання для генерації зображень. Дослідження теоретичних засад створення генеративних зображень, архітектури генеративних моделей штучного інтелекту. Аналіз технологій створення генеративних зображень, методи та засоби проєктування веб-ресурсу. Практична реалізація веб-ресурсу з впровадження генеративних зображень за допомогою моделей машинного навчання.

5. Перелік обов'язкового ілюстративного матеріалу: презентаційний матеріал, цифровий макет проектного веб-ресурсу, перелік та ілюстрування створених генеративних зображень на базі моделі машинного навчання.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1	Ознайомитись з постановкою задач та провести аналіз предметної області	02.10.2023 – 08.10.2023	
2	Проаналізувати теоретичні засади створення генеративних зображень	08.10.2023 – 17.10.2022	
3	Визначити технології створення генеративних зображень, методи та засоби проєктування веб-ресурсу	17.10.2023 – 27.10.2023	
4	Розробити програмний продукт	27.10.2023 – 08.11.2023	
5	Практично реалізувати веб-ресурс з впровадженням генеративних зображень	08.11.2023 – 20.11.2023	
6	Розробити презентаційний матеріал	06.12.2023 – 11.12.2023	

7. Дата видачі завдання «02» жовтня 2023 р.

Керівник кваліфікаційної роботи

_____ (підпис керівника)

Бобарчук О.А.

(П.І.Б.)

Завдання прийняв до виконання

_____ (підпис випускника)

Лебедев В.С.

(П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: «Засоби впровадження генеративних зображень за допомогою машинного навчання моделей (в контексті створення мультимедійного контенту)» містить 91 сторінки, 57 рисунків, 1 таблицю та 50 використаних джерел.

МАКЕТ ВЕБ-РЕСУРСУ, МАШИННЕ НАВЧАННЯ, ГЕНЕРАТИВНІ ЗОБРАЖЕННЯ, DEEP LEARNING, ВЕРСТКА, DIFFUSION MODELS, LLM MODELS

Об'єкт дослідження — моделі машинного навчання на базі Diffusion моделей, аналіз їх різновидів, архітектури, впровадження в мультимедійні контенти, моделі LLM, їх комбінація та приклади використання.

Предмет дослідження — веб-ресурс з застосуванням генеративних зображень за допомогою машинного навчання.

Мета кваліфікаційної роботи — створення проектного веб-ресурсу з впровадженням генеративних зображень за допомогою машинного навчання моделей.

Метод дослідження — теоретичний: аналізування різновидів моделей машинного навчання, таких як Diffusion моделі задля створення теоретичного підґрунтя, яке покликане допомогти обрати необхідну варіацію моделі для створення генеративних зображень машинним навчанням.

Практичний: створення веб-ресурсу з використанням генеративних зображень, підбір та пошук необхідного ПЗ для створення веб-ресурсу, а також створення та впровадження даних зображень на платформу веб-ресурсу.

Практичне значення отриманих результатів: результати виконаної роботи рекомендовано використовувати для подання аудиторії дорослих вікових категорій.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ ГЕНЕРАТИВНИХ ЗОБРАЖЕНЬ, АРХІТЕКТУРА ГЕНЕРАТИВНИХ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ТА ЇХ ПРИНЦИПИ РОБОТИ	11
1.1. Теоретичні засади появи Diffusion моделей генерації зображення	11
1.1.1. Архітектура та структура дифузійних моделей.....	13
1.1.2. Neural Style Transfer (NST).....	17
1.2. Синтез зображень високої роздільної здатності з Latent Diffusion.....	18
1.2.1. Метод підходу Latent Space	21
1.2.2. Новітня дифузійна модель Image-to-Image Diffusion.....	22
1.3. Архітектура GAN як перші кроки до становлення Diffusion моделей.....	24
1.3.1. Перевага GAN у якості методу глибинного навчання	25
1.3.2. Складнощі в тренуванні моделей GAN	26
1.3.3. Короткий огляд процесу створення GAN моделі	28
РОЗДІЛ 2. ТЕХНОЛОГІЯ СТВОРЕННЯ ГЕНЕРАТИВНИХ ЗОБРАЖЕНЬ, МЕТОДИ ТА ЗАСОБИ ПРОЄКТУВАННЯ ВЕБ-РЕСУРСУ.....	44
2.1. Принцип використання AI технології генерації зображення в UI\UX.....	44
2.2. Застосування AI технології генерації зображення як вплив на дизайн.....	47
2.2.1. MidJourney у якості генерації візуальних ідей для веб-дизайну.....	51
2.3. Вибір ШІ інструментарію для реалізації веб-ресурсу.....	55
2.3.1. Вибір між Dall-E 2 та Dall-3 для інтеграції в веб-ресурс	59
2.4. Вибір ПЗ для проєктування веб-ресурсу	63
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-РЕСУРСУ З ВПРОВАДЖЕННЯ ГЕНЕРАТИВНИХ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ.....	67
3.1. Загальні теоретичні відомості по розробці структури веб-ресурсу.....	67
3.1.1. Підготовка загального дизайну проєктного веб-ресурсу	72

3.2. Впровадження додаткових фрейворків у функціонал веб-ресурсу	75
3.3. Фіналізування верстки проєктного веб-ресурсу	78
ВИСНОВКИ	82
СПИСОК БІБЛОГРАФІЧНИХ ПОСИЛАНЬ І ВИКОРИСТАНИХ ДЖЕРЕЛ	85
Додаток А.....	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Diffusion Model — дифузійні моделі є ще однією генеративною моделлю, яка створює високоякісні зображення з текстових описів. Ці моделі тренуються на великих наборах даних зображень і використовують процес розповсюдження для створення ідей, які відповідають вхідному тексту;

Neural Style Transfer (NST) — програма глибокого навчання, яка поєднує вміст одного зображення зі стилем іншого зображення для створення абсолютно нового витвору мистецтва;

Generative Adversarial Networks, або скорочено **GAN**, — архітектура глибокого навчання для навчання потужних моделей генераторів;

LDM — застосування процесів дифузії в латентному просторі замість простору пікселів із включенням семантичного зворотного зв'язку від Transformers;

VAE — варіаційні автокодери.

ВСТУП

Актуальність теми дослідження. Можливо, найважливішим досягненням останнього десятиліття в галузі комп'ютерного бачення та машинного навчання став винахід GAN (Generative Adversarial Networks) — методу, який запровадив можливість мислити за межами того, що вже було присутнє в даних, кроком у цілковито новій галузі, яка тепер називається генеративним моделюванням. Однак, пройшовши через фазу підйому, GAN почали стикатися з плато, коли більшість методів намагалися вирішити деякі вузькі місця, з якими стикалися змагальні методи. Справа не в окремих методах, а в змагальності самої проблеми.

З випуском Dall-E 2, Imagen, Stable Diffusion і Midjourney дифузійні моделі захопили світ штурмом, надихаючи на творчість і розсуваючи межі машинного навчання. Ці моделі можуть генерувати майже нескінченну різноманітність зображень із текстових підказок, у тому числі фотореалістичні, фантастичні, футуристичні та, звичайно, чарівні. Ці можливості переосмислюють, що для людства означає взаємодія з технологіями, надаючи нам здібності створювати майже будь-які зображення, які ми можемо уявити. Але, навіть з їх розширеними можливостями дифузійні моделі мають обмеження. Однак, оскільки ці моделі постійно вдосконалюються або наступна генеративна парадигма бере верх, вони дозволять людству створювати зображення, відео та інші захоплюючі враження простою думкою.

Історично моделі ШІ були зосереджені на сприйнятті та розумінні. Однак великі мовні моделі (LLM, Large Learning Machine), які навчаються на базах даних Інтернет-масштабу із сотнями мільярдів параметрів, тепер розблокували здатність моделі штучного інтелекту створювати контент, подібний до людини.

Моделі вміють читати, писати, кодувати, малювати та творити надійним способом, розширювати людські творчі здібності та підвищувати продуктивність у різних галузях для вирішення найскладніших проблем світу. Програми для цих LLM охоплюють безліч прикладів використання.

Наприклад, система штучного інтелекту може вивчити мову білкових послідовностей, щоб створити життєздатні сполуки, які допоможуть вченим розробити новаторські, рятівні вакцини.

Або комп'ютери можуть допомогти людям робити те, що вони вміють найкраще — бути творчими, спілкуватися та творити. Письменник, який страждає від письменницької блокади, може використовувати велику мовну модель, щоб допомогти розпалити свою творчість. З LLM зазвичай працюють й Diffusion моделі. Дифузійні моделі пропонують кілька переваг перед традиційними генеративними моделями, такими як GAN і VAE.

Об'єкт дослідження — моделі машинного навчання на базі Diffusion моделей, аналіз їх різновидів, архітектури, впровадження в мультимедійні контенти, моделі LLM, їх комбінація та приклади використання.

Предмет дослідження — веб-ресурс з застосуванням генеративних зображень за допомогою машинного навчання.

Метою роботи є створення проектного веб-ресурсу з впровадженням генеративних зображень за допомогою машинного навчання моделей.

Завдання:

- аналізування ймовірних теоретичних засад при дослідженні теми;
- макетування проектного веб-ресурсу завдяки теоретичним основам;
- аналіз існуючих моделей генерації зображень для подальшого впровадження в проектний веб-ресурс;
- обґрунтування вибору програмного забезпечення, що застосоване для створення проекту;
- безпосереднє створення концепту та остаточного дизайну макету веб-ресурсу, що розробляється;
- практична реалізація веб-ресурсу.

Методи дослідження — теоретичний: аналізування різновидів моделей машинного навчання, таких як Diffusion моделі задля створення теоретичного підґрунтя, яке покликане допомогти обрати необхідну варіацію моделі для створення генеративних зображень машинним навчанням.

Практичний: створення веб-ресурсу з використанням генеративних зображень, підбір та пошук необхідного ПЗ для створення веб-ресурсу, а також створення та впровадження даних зображень на платформу веб-ресурсу.

Теоретична база сформована зарубіжними науковими працями в галузі дослідження та створення машинного навчання (deep learning / machine learning).

Структура кваліфікаційної роботи обумовлена її предметом дослідження, метою та поставленими завданнями. Вона складається з коротко викладеної вступної частини, першого розділу (в першому розділі наведена загальна теорія щодо різновидів та структури різноманітних моделей генерації зображень), другого розділу (в цьому розділі представлена розробка дизайн-концепту та вибір програмного забезпечення проектного веб-ресурсу та платформи генерації зображень), третього розділу (в третьому розділі продемонстровано наочне ілюстрування етапів створення композиційно-графічної моделі макету проектного веб-ресурсу та етапів створення й впровадження генеративних зображень), загальних висновків та рекомендацій, списку використаної літератури та джерел.

Технічні та програмні засоби — *SquareSpace, MidJourney, Adobe Photoshop*, текстовий редактор *Microsoft Word*.

Практичне значення отриманих результатів: результати виконаної роботи рекомендовано використовувати для подання аудиторії дорослих вікових категорій.

Особистий внесок: поширення на загал технологій машинного навчання, що покликане спростити життя та допомогти креативним творцям прискорити візуалізацію та поліпшити креативне мислення завдяки генерації зображень.

Апробація результатів

Результати апробацій були оприлюднені шляхом участі у науково-популярних конференціях «Мультимедійні технології в освіті та інших сферах діяльності»: «Потенціал застосування штучного інтелекту в motion дизайні та анімації», а також «Технології композиції та створення інтерактивних електронних публікацій».

РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ ГЕНЕРАТИВНИХ ЗОБРАЖЕНЬ, АРХІТЕКТУРА ГЕНЕРАТИВНИХ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ТА ЇХ ПРИНЦИПИ РОБОТИ

1.1. Теоретичні засади появи Diffusion моделей генерації зображення

Більшість проблем машинного та глибокого навчання, які ви вирішуєте, концептуалізуються на основі генеративної та дискримінаційної моделей. Простіше кажучи, «генеративні моделі» — це статистичні моделі, призначені для «генерування/синтезу даних». Їх робота полягає в тому, щоб «перетворити шум на репрезентативну вибірку даних». Протягом багатьох років світ бачив багато творчих застосувань генеративних моделей. Одним із конкретних застосунків, який більшість пам'ятає, була реклама Cadbury, яка використовувала аудіогенерацію та синхронізацію губ для відображення виразу обличчя та мови різних знаменитостей [1].

Нижче наведені чотири добре відомі моделі створення зображень на основі глибокого навчання:

- Варіаційні автокодери (від англ. Variational Autoencoders, VAE);
- Потоківі моделі (від англ. Flow-based models);
- Генеративні змагальні мережі (Generative Adversarial Networks);
- Дифузія (Diffusion).

Ці моделі спочатку навчаються, щоб вони навчилися моделювати «розподіл даних» (навчальних даних). Після навчання модель знає, як апроксимувати початковий розподіл даних, і може використовувати його для створення нових даних (зображень) за бажанням. Попередником розуміння «варіаційних автокодерів» є «автокодери». Основним завданням автокодерів є стиснення даних. Архітектура автокодерів досить проста на практиці й містить наступні компоненти:

- Кодувальник;
- A Bottleneck (відповідає за стиснення);
- Декодер.

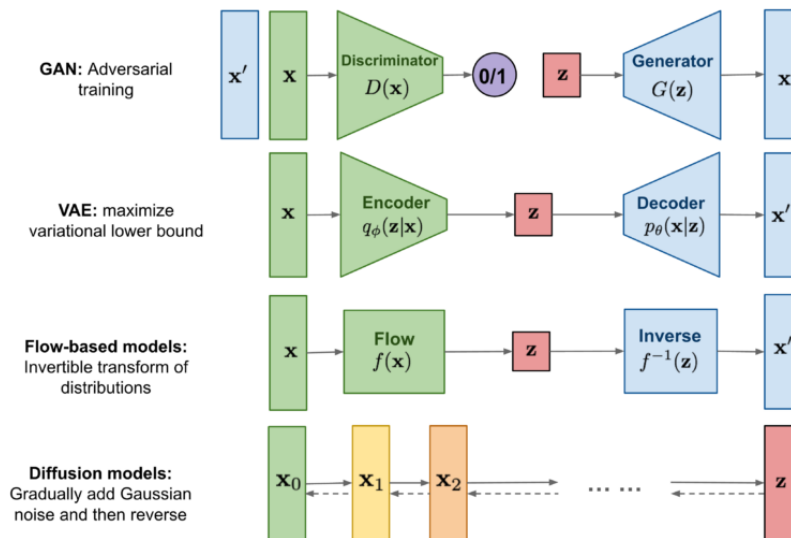


Рис. 1.1. Зображення на високому рівні відображає механізми всіх 4 алгоритмів [1]

У автокодерах розподіл стислих даних/латентів є «необмеженим». Дані стискаються таким чином, щоб помилка реконструкції була мінімальною. Це призводить до великого недоліку, тобто тому, що ми не маємо уявлення/інформації про розподіл прихованого, тому генерувати нові зразки важко [1].

У варіаційних автокодерах це більше не є проблемою. Обмеження додається до рівня вузького місця, тому стиснуті дані кодера повинні якомога точніше імітувати (просто підібраний) розподіл ймовірностей (зазвичай стандартне гауссове значення). Щоб створити нові вибірки, ми можемо просто взяти точку з вибраного розподілу ймовірностей і передати її в декодер. На момент написання даного звіту ще не було зупинки на одній базовій моделі для всіх завдань, пов'язаних із генеративним моделюванням, і це справедливо. Кожна область застосування має свої проблеми, і люди зазвичай використовують різні методи для їх вирішення. У всіх чотирьох перерахованих вище моделях є проблеми, які вдало показано на зображенні нижче (рис. 2).

Створені в 2014 році Ієном Гудфеллоу мережі Generative Adversarial Network (GAN) були майже нормою для створення зразків зображень. Підсумовуючи, нижче наведена таблиця переваг та недоліків моделей створення генеративних зображень, а саме: Variational Autoencoders, Flow-based models, Generative Adversarial Networks та Diffusion [2].

Таблиця 1.1

Переваги та недоліки моделей створення генеративних зображень

	VAE	Flow	GAN	Diffusion
+	Швидка частота дискретизації. Формування різноманітної вибірки.	Швидка частота дискретизації. Формування різноманітної вибірки.	Швидка частота дискретизації. Висока якість генерації зразків.	Висока якість генерації зразків. Формування різноманітної вибірки.
-	Низька якість формування зразків.	Потрібна спеціалізована архітектура, низька якість формування вибірки.	Нестабільне навчання, низька різноманітність генерації вибірки (згортання режиму)	Низька частота дискретизації

1.1.1. Архітектура та структура дифузійних моделей

Стандартна модель дифузії має дві основні області процесів: пряма дифузія та зворотна дифузія. На етапі прямої дифузії зображення спотворюється шляхом поступового введення шуму, доки зображення не стане повністю випадковим шумом. У зворотному процесі серія ланцюгів Маркова використовується для відновлення даних із гаусового шуму шляхом поступового видалення прогнозованого шуму.

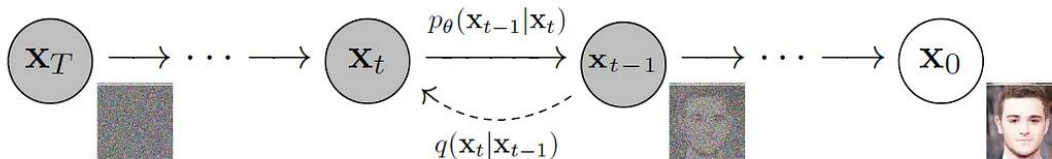


Рис. 1.2. Типовий процес моделі дифузії [7]

Дифузійні моделі нещодавно показали чудову продуктивність у задачах генерації зображень і витіснили продуктивність GAN у кількох завданнях, таких як синтез зображень. Ці моделі також змогли створювати різноманітніші зображення та показали, що вони не страждають від режиму Collapse.

Це пов'язано зі здатністю дифузійних моделей зберігати семантичну структуру даних. Однак ці моделі потребують великих обчислень, а навчання потребує дуже великої пам'яті та вуглецевого сліду, що унеможлиблює для більшості дослідників навіть спробу використання методу.

Це пов'язано з тим, що всі марковські стани повинні постійно бути в пам'яті для передбачення, що означає, що в пам'яті постійно присутні численні екземпляри великих глибоких мереж. Крім того, час навчання таких методів також стає надто великим (наприклад, від днів до місяців), оскільки ці моделі мають тенденцію застрягати в дрібнозернистих непомітних тонкощах даних зображення. Однак слід зазначити, що ця генерація дрібнозернистого зображення також є однією з головних сильних сторін дифузійних моделей, тому їх використання є певним парадоксальним [3]. Інша дуже відома серія методів, що походять із NLP доменів, — це так звані Transformers.

Вони досягли успіху в моделюванні мови та розробці розмовних інструментів ШІ. У додатках для зору Transformers продемонстрували перевагу узагальнення та адаптивності, що робить їх придатними для загального навчання. Вони вловлюють семантичну структуру тексту та навіть зображень краще, ніж інші техніки. Однак Transformers вимагає величезної кількості даних, а також стикається з низьким рівнем продуктивності в багатьох областях зору порівняно з іншими методами.

Ідея, яка використовується в нерівноважній статистичній фізиці, полягає в тому, що є можливість поступово перетворювати один розподіл в інший. У 2015 році Sohl-Dickstein та ін., натхненні цим, створили «імовірнісні моделі дифузії» або, коротше кажучи, «моделі дифузії», спираючись на цю важливу ідею. Нещодавно запропонований метод, який використовує здатність сприйняття GAN, здатність зберігати деталі моделей дифузії та семантичну здатність трансформаторів шляхом об'єднання всіх трьох разом.

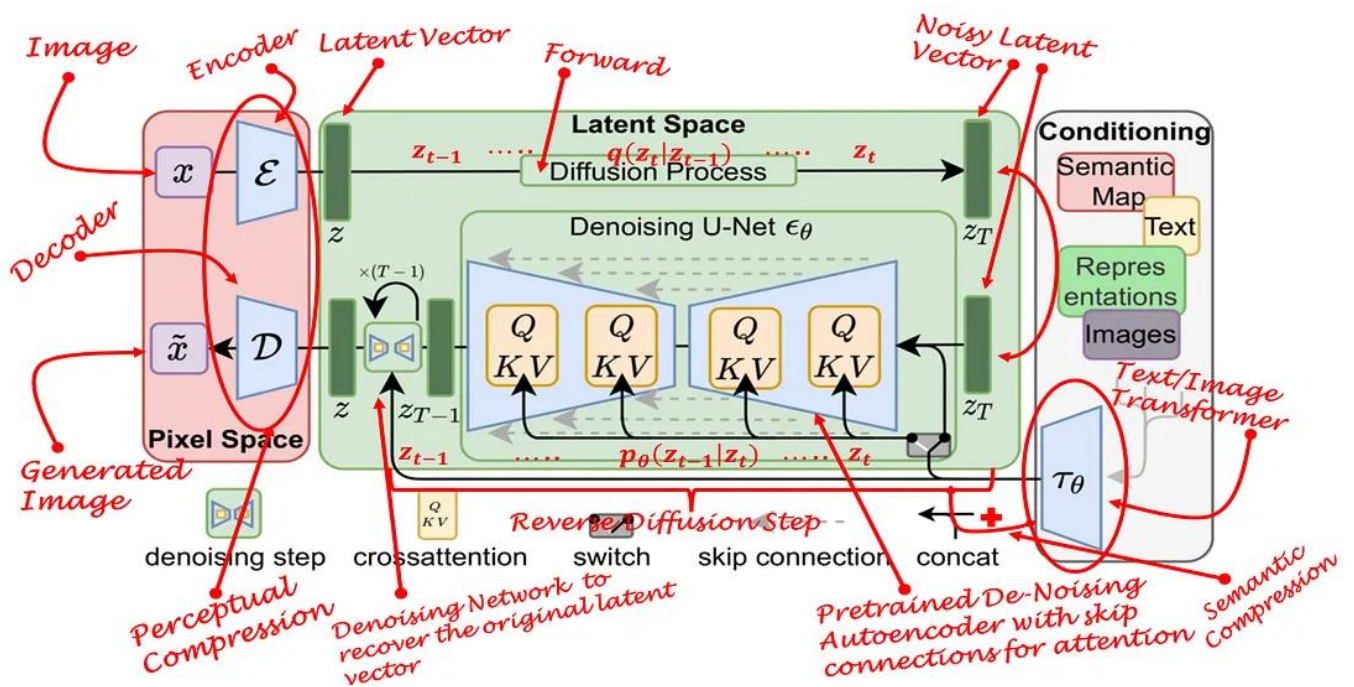


Рис. 1.3. Діаграма Latent Diffusion моделі [4]

LDM зарекомендували себе як більш надійні та ефективні, ніж усі вищезгадані моделі. Порівняно з іншими методами вони не тільки ефективні для пам'яті, але й створюють різноманітні, дуже деталізовані зображення, які зберігають семантичну структуру даних.

LDM — це застосування процесів дифузії в латентному просторі замість простору пікселів із включенням семантичного зворотного зв'язку від Transformers.

Фаза №1: Перцептивне стиснення

Під час фази навчання перцептивного стиснення метод навчання повинен інкапсулювати дані в абстрактне представлення шляхом видалення високочастотних деталей. Цей крок необхідний для створення інваріантного та надійного представлення середовища. Модель GAN добре забезпечують таке перцептивне стиснення. Вони досягають цього шляхом проектування багатовимірних надлишкових даних із простору пікселів у гіперпростір, який називається латентним простором. Прихований вектор у прихованому просторі є стиснутою формою необробленого піксельного зображення, і його можна ефективно використовувати замість необробленого зображення.

Фаза №2: Семантичне стиснення

На другому етапі навчання метод генерації зображень повинен мати можливість охопити семантичну структуру даних. Ця концептуально-семантична структура забезпечує збереження контексту, а також взаємозв'язку різних об'єктів на зображенні. Transformers добре вловлюють семантичну структуру в тексті та зображеннях.

Фаза №3: Втрата сприйняття

Автокодер у LDM — це те, що фіксує перцептивну структуру даних шляхом проектування даних у прихований простір. Для навчання такого автокодувальника автори використовують спеціальну функцію втрат, яка називається «втрата сприйняття» [4–5]. Ця функція втрат гарантує, що реконструкції обмежені колектором зображення та зменшує розмитість, яка інакше була б присутня, коли використовуються втрати піксельного простору (наприклад, втрати L1/L2).

Фаза №4: Втрата дифузії

Дифузійні моделі вивчають розподіл даних шляхом поступового видалення шуму від нормально розподіленої змінної. Іншими словами, DM використовує зворотний ланцюг Маркова довжиною T . Це також означає, що DM можна змодельовувати як серію « T » автокодерів, що зменшують шум, для часових кроків $t = 1, \dots, T$.

«Механізм синтезу на базі “перетворення тексту в зображення” та синтез “зображення в зображення»

Для пояснення механізму використано останню офіційну реалізацію LDM v4 на Python для створення зображень. У синтезі тексту в зображення LDM використовує попередньо навчену модель CLIP [7], яка забезпечує загальне вбудовування на основі трансформатора для кількох модальностей, таких як текст і зображення. Потім вихідні дані моделі трансформатора вводяться в API Python LDM під назвою «дифузори». Є деякі параметри, які також можна налаштувати (наприклад, відсутність кроків розповсюдження, початкове значення, розмір зображення тощо) [6].

1.1.2. Neural Style Transfer (NST)

Neural Style Transfer (NST) — це програма глибокого навчання, яка поєднує вміст одного зображення зі стилем іншого зображення для створення абсолютно нового витвору мистецтва. На високому рівні NST використовує попередньо навчену мережу для аналізу візуальних зображень і використовує додаткові заходи, щоб запозичити стиль з одного зображення та застосувати його до іншого. Це призводить до синтезу нового зображення, яке об'єднує бажані функції.

Процес включає три основні зображення:

Зображення вмісту — це зображення, вміст якого ви бажаєте зберегти.

Зображення стилю — забезпечує художній стиль, який ви хочете накласти на зображення вмісту.

Згенероване зображення — спочатку це може бути випадкове зображення або копія зображення вмісту. З часом це зображення змінюється, щоб поєднати вміст зображення вмісту зі стилем зображення стилю. Це єдина змінна, яку алгоритм фактично змінює під час процесу.

Заглиблюючись у механіку, варто зазначити, що нейронні мережі, які використовуються в NST, мають шари нейронів. Шари, які з'являються першими, можуть розпізнавати краї та кольори, але в міру того, як ви заглиблюєтеся в мережу, шари поєднують ці основні функції, щоб розпізнавати складніші функції, такі як текстури та форми [7].



Рис. 1.4. Процес трансферу нейронного стилю [8]

Втрата вмісту. Якщо необхідно зберегти «вміст» оригінального зображення, це означає, що необхідно, щоб створене зображення мало впізнавані риси вихідного зображення. Втрата вмісту – це міра того, наскільки вміст створеного зображення відрізняється від вмісту вихідного зображення вмісту. NST використовує кілька рівнів нейронних мереж, щоб захопити основні елементи в зображенні та гарантувати, що у згенерованому вмісті ці елементи схожі на елементи вихідного введення.

Втрата стилю. Що стосується стилю, то це більше стосується фактур, кольорів і візерунків в образі. Втрата стилю вимірює відмінності стилю, наприклад, візерунки та текстури в створеному зображенні та зображенні стилю. NST намагається зіставити текстури та візерунки між шарами між зображенням стилю та згенерованим зображенням.

Повна втрата. NST поєднує втрату вмісту та втрату стилю в один показник, який називається загальна втрата. Тут є баланс: якщо ви надто зосереджуєтесь на відповідності вмісту, ви можете втратити стиль, і навпаки. NST дозволяє зважити, наскільки ви дбаєте про вміст і стиль у загальній втраті. Потім він використовує алгоритм оптимізації для зміни пікселів у створеному зображенні, щоб мінімізувати ці загальні втрати. Під час оптимізації згенероване зображення приймає вміст і стиль з інших зображень. Кінцевим результатом є привабливе поєднання обох, часто різоче схоже на витвір мистецтва [8].

1.2. Синтез зображень високої роздільної здатності з Latent Diffusion

Завдяки розкладанню процесу формування зображення на послідовне застосування автокодерів із зменшенням шуму, дифузійні моделі (DM) досягають найсучасніших результатів синтезу даних зображення та інших даних. Крім того, їх формулювання дозволяє створювати керівний механізм для керування процесом генерації зображення без перенавчання. Однак, оскільки ці моделі зазвичай працюють безпосередньо в піксельному просторі, оптимізація потужних DM часто споживає сотні днів GPU, а висновок є дорогим через послідовні оцінки.

Щоб уможливити навчання ДМ на обмежених обчислювальних ресурсах, зберігаючи їх якість і гнучкість, ми застосовуємо їх у латентному просторі потужних попередньо навчених автоматичних кодерів. На відміну від попередньої роботи, навчання дифузійних моделей на такому представленні вперше дозволяє досягти майже оптимальної точки між зменшенням складності та збереженням деталей, значно підвищуючи візуальну точність. Впроваджуючи рівні перехресної уваги в архітектуру моделі, ми перетворюємо дифузійні моделі на потужні та гнучкі генератори для загальних вхідних даних, таких як текст або обмежувальні прямокутники, і синтез з високою роздільною здатністю стає можливим згортковим способом. Наші моделі латентної дифузії (LDM) досягають найсучасніших оцінок для малювання зображень і класового синтезу умовних зображень, а також висококонкурентоспроможної продуктивності в різних завданнях, включаючи безумовне створення зображень, синтез тексту в зображення та супер-роздільною здатністю [9].

Підхід до латентного простору (Latent Space)

Цей підхід починається з аналізу вже навчених моделей дифузії в піксельному просторі: на рис. 6 показано компроміс швидкості спотворення навченої моделі. Як і у випадку з будь-якою моделлю, заснованою на правдоподібності, навчання можна грубо розділити на два етапи: перший — це етап перцептивного стиснення, який видаляє високочастотні деталі, але все ще вивчає невеликі семантичні варіації. На другому етапі фактична генеративна модель вивчає семантичну та концептуальну композицію даних (семантичне стиснення). Таким чином, спостерігається прагнення спочатку знайти перцепційно еквівалентний, але обчислювально більш відповідний простір, у якому ми будемо тренувати дифузійні моделі для синтезу зображень з високою роздільною здатністю. Дотримуючись загальноприйнятої практики, розділяємо навчання на дві окремі фази: по-перше, навчаємо автокодер, який надає маловимірний (і, отже, ефективний) репрезентативний простір, який перцептуально еквівалентний до простору даних. Важливо, що на відміну від попередньої роботи, не потрібно покладатися на надмірне просторове стиснення, оскільки навчаємо ДМ (Дифузійна Модель) у вивченому латентному просторі [10].

Знижена складність також забезпечує ефективну генерацію зображення з прихованого простору за один мережевий прохід. Отримана модель названа класом Latent Diffusion Models (LDM). Помітною перевагою цього підходу є те, що потрібно навчити стадію універсального автокодування лише один раз, і тому можна повторно використовувати її для кількох тренувань DM або для дослідження, можливо, зовсім інших завдань. Це дає змогу ефективно досліджувати велику кількість моделей дифузії для різних завдань із зображенням до зображення та текстом до зображення.

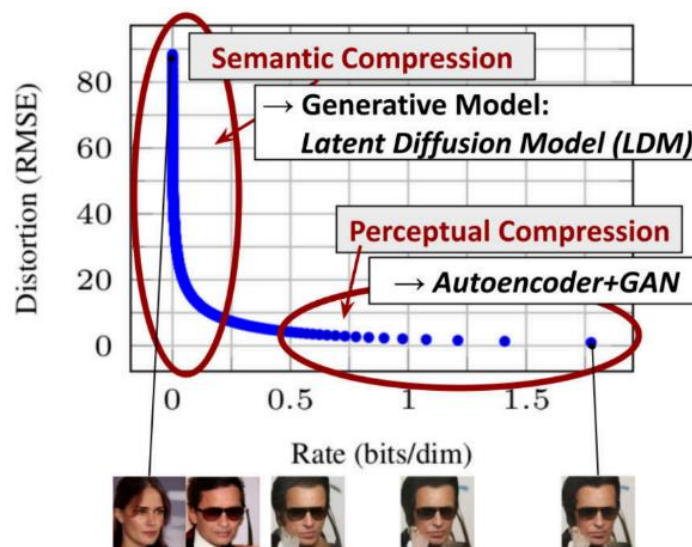


Рис. 1.5. Ілюстрація перцептивного та семантичного стиснення: більшість бітів цифрового зображення відповідають непомітним деталям [11].

Підсумовуючи, робота виконує такі внески: (I) На відміну від чисто трансформаторних підходів, цей метод більш витончено масштабується до даних вищої розмірності і, таким чином, може (a) працювати на рівні стиснення, який забезпечує більш точні та детальні реконструкції, ніж попередні роботи (див. рис. 1) і (b), які можна ефективно застосовувати для синтезу мегапіксельних зображень з високою роздільною здатністю. (II) Таким чином досягається конкурентоспроможна продуктивність в багатьох завданнях (безумовний синтез зображень, малювання, стохастична надроздільна здатність) і наборах даних, значно знижуючи обчислювальні витрати.

Порівняно з підходами дифузії на основі пікселів також значно зменшуються витрати на логічні висновки. (III) Демонструється, що, на відміну від попередньої роботи, яка вивчає як архітектуру кодера/декодера, так і попередні оцінки на основі балів, цей підхід не вимагає делікатного зважування реконструкції та генеративних можливостей [12]. Це забезпечує надзвичайно точні реконструкції та вимагає дуже невеликої регуляризації прихованого простору. (IV) Виявлено, що для щільно обумовлених завдань, таких як надроздільна здатність, замальовування та семантичний синтез, ця модель може бути застосована згортковим способом і відтворювати великі послідовні зображення розміром $\sim 1024^2$ пікселів. (V) Крім того, розробляється механізм кондиціонування загального призначення, заснований на перехресній увазі, що забезпечує мультимодальне навчання. Він використовується для навчання моделей умовного класу, тексту в зображення та макета в зображення.

1.2.1. Метод підходу Latent Space

Щоб знизити обчислювальні вимоги до моделей навчання дифузії для синтезу зображень з високою роздільною здатністю, хоча моделі дифузії дозволяють ігнорувати перцепційно нерелевантні деталі шляхом недобірки відповідних членів втрат, вони все ще вимагають дорогих оцінок функцій у просторі пікселів, що спричиняє величезні вимоги до обчислювальних ресурсів та енергетичних ресурсів. Існує пропозиція обійти цей недолік, запровадивши явне відділення фази стиснення від генеративної фази навчання. Такий підхід пропонує кілька переваг:

(I) Залишаючи простір зображення високої розмірності, ми отримуємо DM, які є набагато ефективнішими з точки зору обчислень, оскільки вибірка виконується в просторі низької розмірності.

(II) Використовується індуктивне зміщення DM, успадковане від їхньої архітектури UNet, що робить їх особливо ефективними для даних з просторовою структурою і зменшує потребу в агресивних рівнях стиснення, що знижують якість, як того вимагали попередні підходи.

(III) Нарешті, отримуємо моделі стиснення загального призначення, чий латентний простір можна використовувати для навчання кількох генеративних моделей і які також можна використовувати для інших додатків, таких як синтез одного зображення, керований CLIP [13].

1.2.2. Новітня дифузійна модель Image-to-Image Diffusion

Більшість дифузійних моделей для малювання та інших лінійних обернених задач адаптували безумовні моделі для використання в умовних задачах. Це має ту перевагу, що потрібно навчити лише одну модель. Однак безумовні завдання часто складніші за умовні. Palette використовується як умовна модель, вибираючи багатозадачне навчання, якщо потрібна одна модель для кількох завдань. Відміна кадрування зображення (також відома як замальовування) вважається більш складним завданням, ніж замальовування, оскільки воно передбачає створення відкритого вмісту з меншим контекстом. Ранні методи ґрунтувалися на пошуку. Зараз переважають методи на основі GAN [Teterwak et al. 2019] [14], але часто є предметно-специфічними.



Рис. 1.6. Враховуючи центральні 256×256 пікселів, екстраполюємо ліворуч і праворуч із кроком 128 пікселів (2×8 додатків із 50% відсіканням палітри), щоб створити остаточну панораму 256×2304 [14].

Дифузійні моделі [Ho et al. 2020; Sohl-Dickstein et al. 2015] [15] перетворюють вибірки зі стандартного розподілу Гаусса на вибірки з розподілу емпіричних даних за допомогою ітераційного процесу усунення шумів.

Моделі умовної дифузії [Chen et al. 2021a; Saharia et al. 2021] зробити процес усунення шумів залежним від вхідного сигналу. Моделі дифузії від зображення до зображення — це моделі умовної дифузії форми $p(y/x)$, де x і y є зображеннями, наприклад, x — зображення у градаціях сірого, а y — кольорове зображення. Ці моделі були застосовані для надроздільності зображень [Nichol and Dhariwal 2021; Сахарія та ін. 2021]. Тут вивчається загальна застосовність моделей дифузії зображення до зображення для широкого кола завдань. Маючи навчальний вихідний образ y , генерується зашумлена версія y_ϵ і навчається нейронна мережа f_θ знешумлювати усе з заданими x та індикатором рівня шуму γ , для якого втрата становить:

$$E_{(x,y)} E_{(\epsilon \sim N(0,1))} E_\gamma \left\| f_\theta \left(x, \frac{\sqrt{\gamma}y + \sqrt{1-\gamma}\epsilon, \gamma \right) - y \right\|_p.$$

Дослідження [Chen та ін. 2021a] та [Saharia et al. 2021] пропонують використовувати норму L1, тобто $p = 1$, тоді як стандартне формулювання базується на звичайній нормі L2 [Ho et al. 2020]. Нижче виконується ретельні абляції та аналізуємо вплив вибору норми. Виявлено, що L1 дає значно меншу різноманітність зразків порівняно з L2. Хоча L1 може бути корисним для зменшення можливих галюцинацій у деяких програмах, тут використовується L2 для точнішого охоплення вихідного розподілу. Архітектура Palette використовує архітектуру U-Net [Ho et al. 2020] [16].



Рис. 1.7. Результати розфарбовування зображень перевірки ImageNet. Базові лінії: † [Guadarrama et al. 2017], ‡ [Kumar et al. 2021], а також власний потужний базовий рівень регресії. Рисунок показує більше зразків [18].

1.3. Архітектура GAN як перші кроки до становлення Diffusion моделей

GAN, як й інші моделі вимагають машинного навчання. Типова проблема машинного навчання залучає використання моделі для створення передбачення, наприклад передбачуване моделювання.

Для цього потрібен навчальний набір даних, який використовується для навчання моделі, що складається з кількох прикладів, які називаються зразками, кожен із вхідними змінними (X) і вихідними мітками класу (y). Модель навчається шляхом показу прикладів вхідних даних, передбачення результатів і коригування моделі, щоб зробити результати більш схожими на очікувані результати.

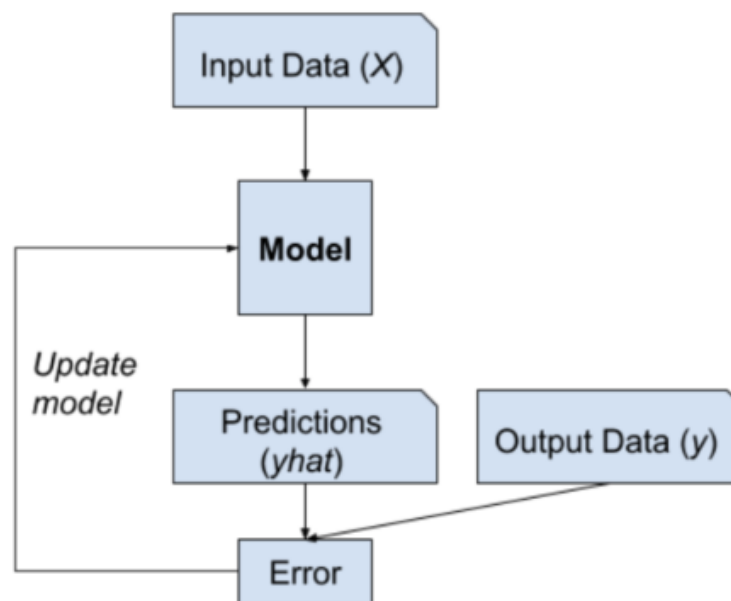


Рис. 1.8. Приклад Supervised навчання [19]

У контрольованому навчанні нас може зацікавити розробка моделі для прогнозування мітки класу на прикладі вхідних змінних. Це завдання прогнозного моделювання називається класифікацією. Класифікацію також традиційно називають дискримінаційним моделюванням. Це пояснюється тим, що модель повинна розрізняти приклади вхідних змінних між класами; він повинен вибрати або прийняти рішення щодо того, до якого класу належить даний приклад.

Як альтернатива, неконтрольовані моделі, які підсумовують розподіл вхідних змінних, можна використовувати для створення або створення нових прикладів у вхідному розподілі.

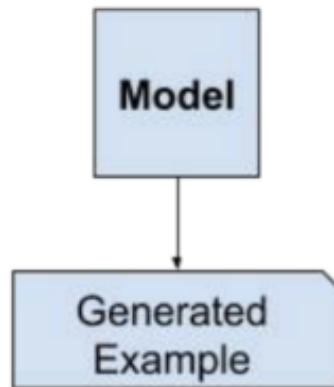


Рис. 1.9. Приклад Generative моделювання [19]

1.3.1. Перевага GAN у якості методу глибинного навчання

Одним із багатьох значних досягнень у використанні методів глибокого навчання в таких сферах, як комп'ютерний зір, є техніка, яка називається доповненням даних. Розширення даних призводить до кращих продуктивних моделей, як підвищуючи навички моделі, так і забезпечуючи регулюючий ефект, зменшуючи помилку узагальнення. Він працює шляхом створення нових, штучних, але правдоподібних прикладів із вхідної проблемної області, на якій навчається модель. Техніка є примітивною у випадку даних зображень, включаючи кадрування, перевертання, масштабування та інші прості трансформації існуючих зображень у навчальному наборі даних. Успішне генеративне моделювання забезпечує альтернативний і потенційно більш предметно-спеціальний підхід до збільшення даних. У складних областях або областях з обмеженою кількістю даних генеративне моделювання забезпечує шлях до більшого навчання моделюванню. GAN досягли значного успіху в цьому випадку використання в таких сферах, як глибоке підсилювальне навчання. Є багато дослідницьких причин, чому GAN цікаві, важливі та потребують подальшого вивчення.

Ієн Гудфеллоу описує деякі з них у своїй доповіді на конференції 2016 року та пов'язаному з нею технічному звіті під назвою «NIPS 2016 Tutorial: Generative Adversarial Networks» [8]. Серед цих причин він виділяє GAN для забезпечення мультимодальних виходів або кількох правдоподібних відповідей.

Надроздільна здатність зображення: можливість генерувати версії вхідних зображень із високою роздільною здатністю;

Створення мистецтва: здатність створювати нові та художні образи, ескізи, живопис тощо;

Переклад із зображення в зображення: можливість перекладати фотознімки в різних доменах, як-от день у ніч, літо в зиму тощо.

Можливо, найбільш вагомою причиною широкого вивчення, розробки та використання GAN є їхній успіх. GAN вдалося створити настільки реалістичні фотографії, що люди не можуть визначити, що на них зображені об'єкти, сцени та люди, які не існують у реальному житті.



Рис. 1.10. Приклад прогресу у можливостях GAN з 2014 до 2017 років [20].

1.3.2. Складнощі в тренуванні моделей GAN

GAN важко піддаються навчанню. Причина, чому їх важко навчити, полягає в тому, що і модель генератора, і модель дискримінатора навчаються одночасно в грі. Це означає, що вдосконалення однієї моделі відбуваються разом з іншою. Метою навчання двох моделей є знаходження точки рівноваги між двома конкуруючими кутами.

Це також означає, що кожного разу, коли параметри однієї з моделей оновлюються, характер оптимізаційної задачі, яка вирішується, змінюється.

З точки зору нейронних мереж, технічна проблема одночасного навчання двох конкуруючих нейронних мереж полягає в тому, що вони можуть не зійтися. Замість конвергенції GAN можуть страждати від одного з невеликої кількості режимів відмови. Поширеним режимом відмови є те, що замість пошуку точки рівноваги генератор коливається між генеруванням конкретних прикладів у домені. Мабуть, найскладнішою помилкою моделі є випадок, коли кілька входів у генератор призводять до генерації одного і того ж виходу. Це називається згортанням режиму та може являти собою одну з найскладніших проблем під час навчання GAN. Немає хороших об'єктивних показників для оцінки того, чи добре працює GAN під час навчання. наприклад перегляду втрат недостатньо. Натомість найкращим підходом є візуальна перевірка згенерованих прикладів і використання суб'єктивної оцінки [21].

Можливо, одним із найважливіших кроків у розробці та навчанні стабільних режимів GAN стала стаття 2015 року Алека Редфорда та ін. під назвою Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. У статті вони описують глибокий згортковий GAN, або DCGAN.

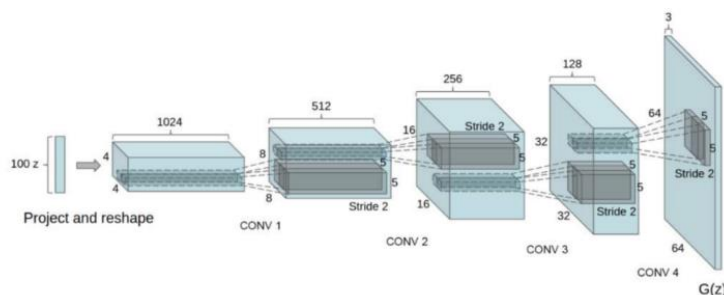


Рис. 1.11. Приклад архітектури Generator моделі для DCGAN [22]

Висновки в цьому документі були важко зароблені, розроблені після численних емпіричних проб і помилок з різними архітектурами моделей, конфігураціями та схемами навчання. Їхній підхід залишається дуже рекомендованим як відправна точка при розробці нових GAN, принаймні для завдань, заснованих на синтезі зображень.

1.3.3. Короткий огляд процесу створення GAN моделі

Generative Adversarial Networks, або скорочено GAN, — це архітектура глибокого навчання для навчання потужних моделей генераторів. Модель генератора здатна генерувати нові штучні зразки, які ймовірно могли виникнути з існуючого розподілу зразків. GAN скомпрометовані як для моделей генератора, так і для дискримінатора. Generator відповідає за створення нових зразків із домену, а дискримінатор відповідає за класифікацію зразків справжніми чи підробленими (згенерованими). Це означає, що генератор насправді ніколи не бачить приклади з домену та адаптується на основі того, наскільки добре працює дискримінатор. Це складний тип моделі як для розуміння, так і для навчання. Одним із підходів до кращого розуміння природи моделей GAN і того, як їх можна навчити, є розробка моделі з нуля для дуже простого завдання. Простим завданням, яке забезпечує хороший контекст для розробки простого GAN з нуля, є одновимірна функція. Це пояснюється тим, що як реальні, так і згенеровані зразки можна побудувати та візуально перевірити, щоб отримати уявлення про те, що було вивчено. Проста функція також не потребує складних моделей нейронної мережі, тобто можна легко зрозуміти конкретні моделі генератора та дискримінатора [23].

1. Обрати одно-просторову функцію

Першим кроком є вибір одновимірної функції для моделювання:

$$y = f(x),$$

де x — вхідні значення;

y — вихідні значення функції.

Зокрема, необхідна функція, яку можна легко зрозуміти та побудувати. Це допоможе як у встановленні очікувань щодо того, що має генерувати модель, так і у використанні візуального огляду згенерованих прикладів, щоб отримати уявлення про їхню якість. Надалі використовуватиметься проста функція x -квадрат; тобто функція поверне квадрат вхідних даних. Запуск прикладу обчислює вихідне значення для кожного вхідного значення та створює графік вхідних і вихідних значень.

Помітно, що значення, далекі від 0,0, призводять до більших вихідних значень, тоді як значення, близькі до нуля, призводять до менших вихідних значень.

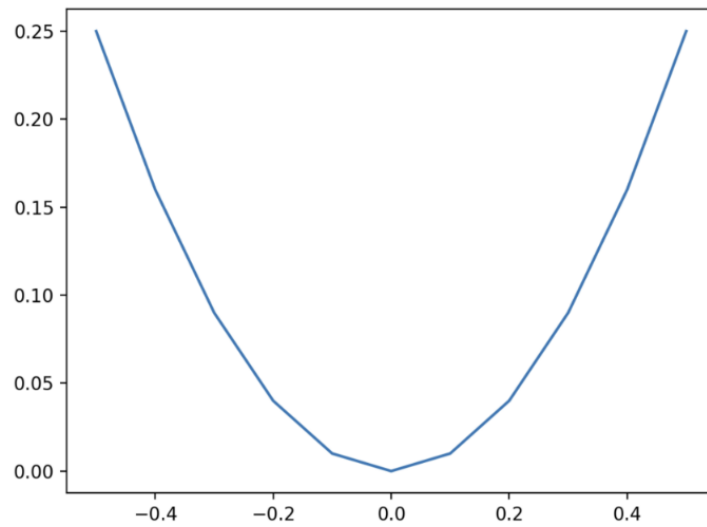


Рис. 1.12. Графік входів і виходів для функції X-квадрат [24]

Можна генерувати випадкові вибірки або точки з функції. Цього можна досягти шляхом генерації випадкових значень від -0,5 до 0,5.

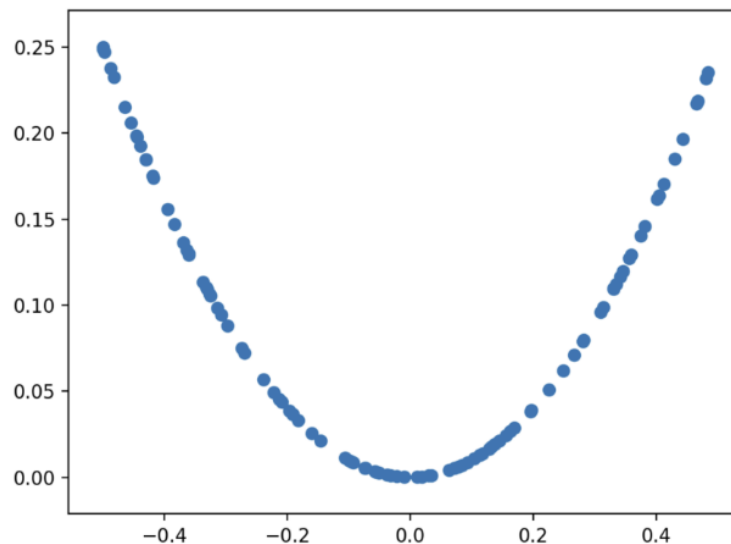


Рис. 1.13. Діаграма випадково згенерованої вибірки вхідних даних і обчислених виходів для функції x-квадрат [24]

Побудова цих зразків за допомогою діаграми розсіювання покаже той самий U-подібний графік, хоча він і складається з окремих випадкових значень між 0 і 1, а потім переходить до діапазону -0,5 і 0,5. Потім обчислюється вихідне значення для кожного випадково згенерованого вхідного значення та об'єднуємо масиви в єдиний масив NumPy з n рядків (100) і двох стовпців.

2. Визначення моделі дискримінатора (Discriminator)

Наступним кроком є визначення моделі дискримінатора. Модель повинна взяти зразок із нашої задачі, такий як вектор із двома елементами, і вивести класифікаційний прогноз щодо того, чи є зразок справжнім чи фальшивим. Це проблема бінарної класифікації.

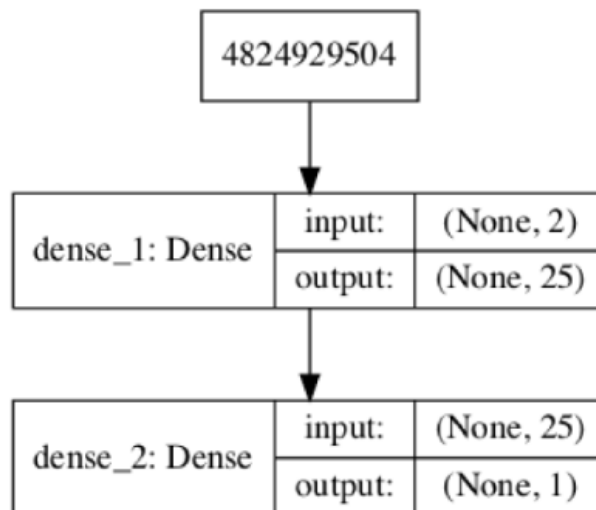


Рис. 1.14. Графік Discriminator моделі в GAN [25]

3. Визначення моделі генератора (Generator)

Наступним кроком є визначення моделі генератора. Модель генератора приймає як вхідні дані точку з прихованого простору та генерує нову вибірку, наприклад, вектор із вхідними та вихідними елементами нашої функції, наприклад, x і x -квадрат. Латентна змінна — це прихована або неспостережувана змінна, а латентний простір — це багатовимірний векторний простір цих змінних. Можливо визначити розмір прихованого простору для нашої проблеми та форму чи розподіл змінних у прихованому просторі [26].

Це пояснюється тим, що латентний простір не має сенсу, поки модель генератора не почне призначати значення точкам у зовнішньому просторі, коли вона навчається. Після навчання точки в прихованому просторі будуть відповідати точкам у зовнішньому просторі, напр. у просторі згенерованих зразків. Ми визначимо малий латентний простір із п'яти вимірів і використаємо стандартний підхід у літературі GAN щодо використання розподілу Гауса для кожної змінної в латентному просторі.

Також створюється графік моделі, і ми бачимо, що модель очікує п'ятиелементну точку з прихованого простору як вхідні дані та передбачає двоелементний вектор як вихідні дані [27].

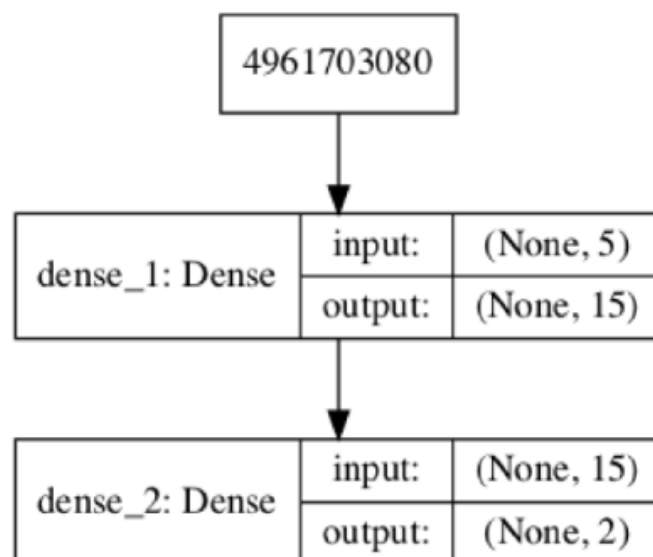


Рис. 1.15. Графік Generator моделі в GAN

Запуск прикладу генерує 100 випадкових точок із латентного простору, використовує це як вхідні дані для генератора 100 фальшивих зразків із нашої одновимірної функціональної області. Оскільки генератор не був навчений, згенеровані точки є повним сміттям, але ми можемо уявити, що якщо модель навчена, ці точки повільно почнуть нагадувати цільову функцію та її U-подібну форму.

Видно, що модель приймає як вхідні дані випадковий п'ятиелементний вектор із латентного простору та виводить двоелементний вектор для нашої одновимірної функції.

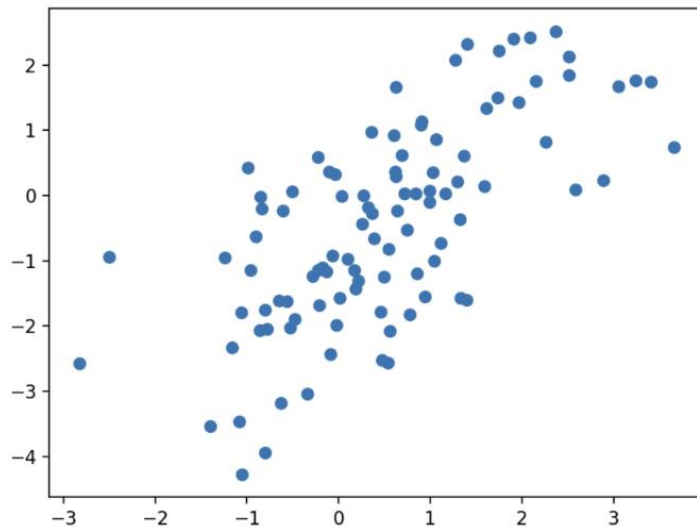


Рис. 1.16. Діаграма розсіювання фальшивих зразків, передбачених моделлю генератора

3. Визначення моделі генератора (Generator)

Як правило, не існує об'єктивних способів оцінити продуктивність моделі GAN. У цьому конкретному випадку можна розробити об'єктивну міру для згенерованих зразків, оскільки відома справжня базова вхідна область.

4. Повний приклад навчання GAN

Повний приклад наведено нижче:

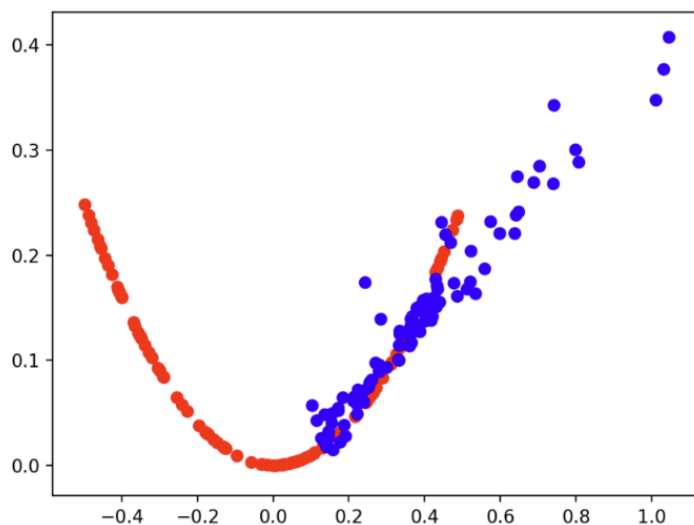


Рис. 1.17. Діаграма розсіювання реальних і згенерованих прикладів

Другий графік показує справжній (червоний) проти фальшивого (синій) після 10000 ітерацій. Тут зрозуміло, що модель генератора виконує розумну роботу з генерування правдоподібних зразків із вхідними значеннями в правій області між (-0,5 і 0,5) і вихідними значеннями, які показують співвідношення х-квадрат [28].

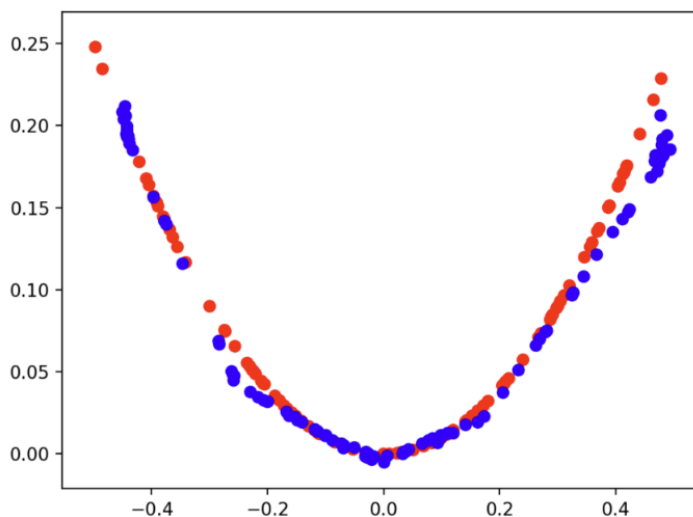


Рис. 1.18. Діаграма розсіювання реальних і згенерованих прикладів для цільової функції після 10000 ітерацій

Тренування моделі генератора

Вагові коефіцієнти в моделі генератора оновлюються на основі продуктивності моделі дискримінатора. Коли дискримінатор добре виявляє підроблені зразки, генератор оновлюється більше, а коли модель дискримінатора є відносно поганою або заплутаною під час виявлення підроблених зразків, модель генератора оновлюється менше. Це визначає нульову суму або змагальність між цими двома моделями. Може бути багато способів реалізувати це за допомогою Keras API, але, мабуть, найпростішим підходом є створення нової моделі, яка поєднує моделі генератора та дискримінатора. Зокрема, можна визначити нову модель GAN, яка об'єднує генератор і дискримінатор таким чином, що генератор отримує як вхід випадкові точки в латентному просторі та генерує зразки, які безпосередньо подаються в модель дискримінатора, класифікуються, а вихідні дані цієї більшої моделі можуть бути використані для оновлення ваг моделі генератора [29].

Щоб було зрозуміло, тут мова йде не про нову третю модель, лише про нову логічну модель, яка використовує вже визначені шари та ваги з автономних моделей генератора та дискримінатора. Тільки дискримінатор займається розрізненням справжніх і фальшивих прикладів, тому модель дискримінатора можна навчити окремо на прикладах кожного, як ми робили в розділі про модель дискримінатора вище.

Модель генератора стосується лише продуктивності дискримінатора на підроблених прикладах. Тому ми позначимо всі шари в дискримінаторі як такі, що не підлягають навчанню, якщо він є частиною моделі GAN, щоб їх не можна було оновлювати та перенавчати на підроблених прикладах. Під час навчання генератора через цю логічну модель GAN є ще одна важлива зміна. Необхідно, щоб дискримінатор думав, що зразки, виведені генератором, справжні, а не підроблені. Тому, коли генератор навчається як частина моделі GAN, ми будемо позначати згенеровані зразки як реальні (клас = 1).

В чому полягає причина? Можна уявити, що потім дискримінатор класифікує згенеровані зразки як нереальні (клас = 0) або низьку ймовірність бути реальними (0,3 або 0,5). Процес зворотного розповсюдження, який використовується для оновлення вагових коефіцієнтів моделі, розцінить це як велику помилку та оновить вагові коефіцієнти моделі, щоб виправити цю помилку, у свою чергу покращуючи генератор у створенні хороших підроблених зразків [30]. Вхідні дані: точка в латентному просторі, напр. 100-елементний вектор випадкових чисел Гауса. Результати: двійкова класифікація, ймовірність того, що зразок справжній (або підроблений).

Наведена нижче функція *define_gan()* приймає як аргументи вже визначені моделі генератора та дискримінатора та створює нову логічну третю модель, яка включає ці дві моделі. Вагові коефіцієнти в дискримінаторі позначено як такі, що їх неможливо навчити, що впливає лише на вагові коефіцієнти, як їх бачить модель GAN, а не автономна модель дискримінатора. Потім модель GAN використовує ту саму двійкову функцію перехресних ентропійних втрат як дискримінатор і ефективну версію Адама стохастичного градієнтного спуску зі швидкістю навчання 0,0002 та

імпульсом 0,5, рекомендованим під час навчання глибокого згорткового GANS. Зробити дискримінатор недоступним для навчання — це хитрий трюк у Keras API.

Властивість `trainable` впливає на модель після її компіляції. Модель дискримінатора було скомпільовано з шарами, які можна навчати, тому ваги моделі в цих шарах буде оновлено, коли автономна модель буде оновлено через виклики функції `train_on_batch()`. Потім модель дискримінатора було позначено як непридатну для навчання, додано до моделі GAN і скомпільовано.

У цій моделі вагові коефіцієнти моделі дискримінатора не піддаються навчанню та не можуть бути змінені, коли модель GAN оновлюється через виклики функції `train_on_batch()`. Ця зміна властивості, що піддається навчанню, не впливає на навчання моделі автономного дискримінатора. Повний приклад створення дискримінатора, генератора та складеної моделі наведено нижче:

```
# demonstrate creating the three models in the gan
from keras.optimizers import Adam
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Reshape
from keras.layers import Flatten
from keras.layers import Conv2D
from keras.layers import Conv2DTranspose
from keras.layers import LeakyReLU
from keras.layers import Dropout
from keras.utils.vis_utils import plot_model
# define the standalone discriminator model
def define_discriminator(in_shape=(32,32,3)):
    model = Sequential()
    # normal
    model.add(Conv2D(64, (3,3), padding= 'same', input_shape=in_shape))
    model.add(LeakyReLU(alpha=0.2))
```

Навчання складеної моделі передбачає генерування пакетної вартості балів у прихованому просторі за допомогою функції *generate_latent_points()* у попередньому розділі та мітки *class = 1* і виклик функції *train_on_batch()*. Наведена нижче функція *train_gan()* демонструє це, хоча вона досить проста, оскільки лише генератор оновлюватиметься кожної епохи, залишаючи дискримінатор із вагами моделі за замовчуванням. Замість цього потрібно спочатку оновити модель дискримінатора за допомогою реальних і підроблених зразків, а потім оновити генератор за допомогою складеної моделі. Для цього потрібно поєднати елементи функції *train_discriminator()*, визначеної в розділі дискримінатора вище, і функції *train_gan()*, визначеної вище.

Є кілька речей, на які слід звернути увагу в цій функції навчання моделі. По-перше, кількість пакетів у межах епохи визначається тим, скільки разів розмір пакету ділиться на навчальний набір даних. Маємо розмір набору даних 50 тисяч зразків і розмір пакета 128, тож із округленням у меншу сторону виходить $\frac{5000}{128}$ або 390 пакетів на епоху. Модель дискримінатора оновлюється двічі на партію, один раз 128 із справжніми зразками та один раз із підробленими зразками, що є найкращою практикою на відміну від об'єднання зразків і виконання одного оновлення. Нарешті, ми повідомляємо про втрати кожної партії. Дуже важливо стежити за втратою партій. Причиною цього є те, що збій у втратах дискримінатора вказує на те, що модель генератора почала генерувати непотрібні приклади, які дискримінатор може легко розрізнити. Слідкуйте за втратою дискримінатора та очікуйте, що вона коливатиметься приблизно від 0,5 до 0,8 на партію. Втрати генератора є менш критичними і можуть коливатися від 0,5 до 2 або вище. Розумний програміст може навіть спробувати виявити аварійну втрату дискримінатора, зупинити, а потім перезапустити процес навчання [30].

Оцінка продуктивності моделі GAN

Як правило, не існує об'єктивних способів оцінити продуктивність моделі GAN. Не існує можливості розрахувати цю об'єктивну оцінку помилок для створених зображень. Натомість якість зображень повинна суб'єктивно оцінюватися людиною-оператором. Це означає, що ми не можемо знати, коли припинити навчання, не переглянувши приклади згенерованих зображень. У свою чергу, змагальний характер

процесу навчання означає, що генератор змінюється після кожної партії, а це означає, що як тільки можна створити достатньо хороші зображення, суб'єктивна якість зображень може почати змінюватися, покращуватися або навіть погіршуватися з наступними оновленнями. Є три способи вирішення цієї складної навчальної ситуації.

1. Періодично оцінювати точність класифікації дискримінатора на справжніх і підроблених зображеннях;

2. Періодично створювати багато зображень і зберігайте їх у файлі для суб'єктивного перегляду;

3. Періодично зберігати модель генератора.

Усі три ці дії можна виконувати одночасно для даної епохи навчання, наприклад кожні 10 епох навчання. Результатом буде збережена модель генератора, для якої маємо можливість суб'єктивно оцінити якість її виходу та об'єктивно знати, наскільки добре дискримінатор був обдурений під час збереження моделі. Навчання GAN протягом багатьох епох, наприклад, сотні чи тисячі епох, призведе до багатьох знімків моделі, які можна перевірити, і з яких можна вибрати конкретні результати та моделі для подальшого використання. По-перше, можна визначити функцію під назвою *summarize_performance()*, яка підсумовуватиме продуктивність моделі дискримінатора. Це робиться шляхом отримання зразка справжніх зображень CIFAR-10, а також генерації такої ж кількості підроблених зображень CIFAR-10 за допомогою моделі генератора, а потім оцінювання точності класифікації.

```
# evaluate the discriminator, plot generated images, save generator model
def summarize_performance(epoch, g_model, d_model, dataset, latent_dim,
n_samples=150):
    # prepare real samples
    X_real, y_real = generate_real_samples(dataset, n_samples)
    # evaluate discriminator on real examples
    _, acc_real = d_model.evaluate(X_real, y_real, verbose=0)
    # prepare fake examples
    x_fake, y_fake = generate_fake_samples(g_model, latent_dim, n_samples)
    # evaluate discriminator on fake examples
```

Повний приклад GAN для CIFAR-10

Виконання прикладу може зайняти багато годин на апаратному забезпеченні ЦП. Рекомендується запуснути приклад на обладнанні GPU, якщо це можливо. Якщо вам потрібна допомога, ви можете швидко почати, використовуючи примірник AWS EC2 для навчання моделі. Див. інструкції в Додатку С. Вибрана конфігурація забезпечує стабільне навчання як генеративної, так і дискримінаційної моделей. Продуктивність моделі повідомляється в кожній партії, включаючи втрату як дискримінаційної (d), так і генеративної (g) моделей

Конкретні результати можуть відрізнитися враховуючи стохастичний характер алгоритму навчання. Варто спробувати запуснути приклад кілька разів і порівняти середню продуктивність. У цьому випадку втрата залишається стабільною протягом усього тренування. Втрати дискримінатора на реальних і згенерованих прикладах становлять близько 0,5, тоді як втрати для генератора, навченого через дискримінатор, становлять близько 1,5 протягом більшої частини процесу навчання.

Після вибору остаточної моделі генератора її можна використовувати окремо для програми. Це передбачає спочатку завантаження моделі з файлу, а потім її використання для створення зображень. Для створення кожного зображення потрібна точка в латентному просторі як вхідні дані. Повний приклад завантаження збереженої моделі та створення зображень наведено нижче. У цьому випадку використовуватиметься модель, збережену після 200 епох навчання.

Векторна арифметика в латентному просторі

Модель генератора в архітектурі GAN приймає точку з латентного простору як вхідні дані та створює новий образ. Прихований простір сам по собі не має сенсу. Як правило, це 100-вимірне зображення гіперсфера з кожною змінною, взятою з розподілу Гауса із середнім значенням нуль і стандартне відхилення одиниці. Під час навчання генератор вчиться відображати точки на латентний простір з конкретними вихідними зображеннями, і це відображення буде різним кожного разу, коли модель навчається. Прихований простір має структуру, коли його інтерпретує модель генератора, і це структуру можна запитувати та здійснювати навігацію для заданої моделі.

Як правило, нові зображення генеруються з використанням випадкових точок у прихованому просторі. Якщо зробити крок далі, точки в латентному просторі можуть бути створені (наприклад, усі 0s, усі 0,5 або всі 1) і використані як вхідні дані або запит для створення конкретного зображення. Ряд точок може бути створений на лінійному шляху між двома точками в прихованому просторі, наприклад, два згенерованих зображення. Ці точки можна використовувати для створення серії зображень, які показують перехід між двома згенерованими зображеннями. Нарешті, точки в прихованому просторі можна зберігати та використовувати в простій векторній арифметиці для створення нових точок у прихованому просторі, які, у свою чергу, можна використовувати для створення зображень. Це цікава ідея, оскільки вона дозволяє інтуїтивно зрозуміле та цілеспрямоване створення зображень. Важлива стаття 2015 року Алека Редфорда та ін. під назвою *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks* представила стабільну конфігурацію моделі для навчання моделей глибокої згорткової нейронної мережі як частину архітектури GAN. У статті автори досліджували прихований простір для GAN, що вписується в низку різних наборів навчальних даних, особливо в набір даних обличч знаменитостей. Вони продемонстрували дві цікаві сторони. Першою була векторна арифметика з гранями. Наприклад, обличчя усміхненої жінки мінус обличчя нейтральної жінки плюс обличчя нейтрального чоловіка призвело до обличчя усміхненого чоловіка [31].

Першим кроком є вибір набору даних обличч. Використовуватиметься великомасштабний набір атрибутів CelebFaces, який називається CelebA. Цей набір даних розроблено та опубліковано Ziwei Liu та ін. за їх статтю 2015 року під назвою «Від реакції частин обличчя до виявлення обличчя: підхід до глибокого навчання» [31]. Набір даних надає близько 200 000 фотографій обличч знаменитостей разом із анотаціями до того, що відображається на цих фотографіях, як-от окуляри, форма обличчя, головні убори, тип волосся тощо. У рамках набору даних автори надають версію кожної фотографії, центром якої є обличчя та обрізаний до портрета з різними розмірами приблизно 150 пікселів у ширину та 200 пікселів у висоту. Використовуватиметься це як основу для розробки нашої моделі GAN.

Набір даних можна легко завантажити з веб-сторінки Kaggle. Знадобиться обліковий запис у Kaggle. Набір даних CelebFaces Attributes (CelebA) [31].

Зокрема, слід завантажити файл `img_align_celeba.zip` розміром приблизно 1,3 гігабайта. Для цього треба клацнути назву файлу на веб-сайті Kaggle. Завантаження може тривати деякий час залежно від швидкості вашого інтернет-з'єднання. Після завантаження розпакуйте архів. Опісля цього буде створено новий каталог під назвою `img_align_celeba/`, який містить усі зображення з іменами файлів на зразок `202599.jpg` і `202598.jpg`. Далі розглянемо підготовку необроблених зображень для моделювання.

Прогресивне зростання GAN

Генеративні змагальні мережі, або скорочено GAN, є ефективним підходом для навчання моделей глибокої згорткової нейронної мережі для створення синтетичних зображень. Навчання моделі GAN включає дві моделі: генератор, який використовується для виведення синтетичних зображень, і модель дискримінатора, яка використовується для класифікації зображень як справжніх або підроблених, яка використовується для навчання моделі генератора. Дві моделі тренуються разом у змагальній манері, шукаючи рівноваги. Порівняно з іншими підходами, вони є швидкими та забезпечують чіткі зображення. Проблема з GAN полягає в тому, що вони обмежені невеликими розмірами набору даних, часто кількома сотнями пікселів і часто квадратними зображеннями розміром менше 100 пікселів.

«GAN створюють чіткі зображення, хоча й лише з досить малою роздільною здатністю та з дещо обмеженими варіаціями, і навчання продовжує бути нестабільним, незважаючи на останні досягнення» [32]. «Вважається, що створення зображень із високою роздільною здатністю є складним завданням для моделей GAN, оскільки генератор повинен навчитися виводити як велику структуру, так і дрібні деталі одночасно. Висока роздільна здатність дозволяє дискримінатору легко помітити будь-які проблеми в дрібних деталях згенерованих зображень, а процес навчання буде невдалим» [33]. «Генерація зображень високої роздільної здатності складна, оскільки вища роздільна здатність полегшує різницю згенерованих зображень від навчальних зображень» [34].

Великі зображення, такі як 1024-піксельні квадратні зображення, також вимагають значно більше пам'яті, яка порівняно з основною пам'яттю є відносно обмеженою на сучасному апаратному забезпеченні GPU. Таким чином, розмір партії, який визначає кількість зображень, що використовуються для оновлення вагових коефіцієнтів моделі, на кожній ітерації навчання має бути зменшено, щоб гарантувати, що великі зображення вміщуються в пам'ять. Це, у свою чергу, вносить додаткову нестабільність у тренувальний процес. Велика роздільна здатність також вимагає використання менших міні-серій через обмеження пам'яті, що ще більше погіршує стабільність навчання. Крім того, навчання моделей GAN залишається нестабільним, навіть за наявності набору емпіричних методів, призначених для підвищення стабільності процесу навчання моделі.

Рішення проблеми навчання стабільних моделей GAN для великих зображень полягає в поступовому збільшенні кількості шарів під час процесу навчання. Цей підхід називається прогресивним зростанням GAN, прогресивним GAN або скорочено PGGAN. Підхід був запропонований Тего Karras та ін. від Nvidia в документі 2017 року під назвою «Поступове зростання GAN для покращення якості, стабільності та варіації», представленою на конференції ICLR у 2018 році [34].

Основний внесок — це методологія навчання GAN, де ми починаємо із зображень із низькою роздільною здатністю, а потім поступово збільшуємо роздільну здатність, додаючи шари до мереж. Поступове зростання GAN для покращення якості, стабільності та варіації, 2017. Прогресивне зростання GAN передбачає використання моделі генератора та дискримінатора з однаковою загальною структурою та починаючи з дуже маленьких зображень, наприклад 4 x 4 пікселя. Під час навчання нові блоки згорткових шарів систематично додаються як до моделі генератора, так і до моделей дискримінатора. Прогресивне зростання GAN вимагає, щоб потужність як генератора, так і моделі дискримінатора була розширена шляхом додавання шарів під час процесу навчання. Це дуже схоже на пошаровий процес навчання, який був звичайним для розробки нейронних мереж глибокого навчання до розробки ReLU та Batch Normalization.

Поетапне введення нового блоку шарів передбачає використання з'єднання пропуску для підключення нового блоку до входу дискримінатора або виходу генератора та додавання його до існуючого вхідного або вихідного рівня за допомогою зважування. Зважування контролює вплив нового блоку та досягається за допомогою параметра альфа (α), який починається з нуля або дуже малого числа та лінійно збільшується до 1,0 протягом навчальних ітерацій.

Відсутність контролю над синтезованими зображеннями

Генеративні змагальні мережі ефективні для створення високоякісних синтетичних зображень із великою роздільною здатністю. Модель генератора приймає як вхідні дані точку з латентного простору та генерує зображення. Ця модель навчається за допомогою другої моделі, яка називається дискримінатором, яка вчиться відрізняти реальні зображення з навчального набору даних від підроблених зображень, згенерованих моделлю генератора. Таким чином, дві моделі змагаються у змагальній грі та знаходять баланс або рівновагу під час тренувального процесу. Чимало вдосконалень архітектури GAN було досягнуто завдяки вдосконаленню моделі дискримінатора. Ці зміни мотивовані ідеєю, що краща модель дискримінатора, у свою чергу, призведе до створення більш реалістичних синтетичних зображень. Таким чином, генератор був дещо занедбаний і залишається чорним ящиком. Наприклад, джерело випадковості, що використовується при створенні синтетичних зображень, недостатньо зрозуміле, включаючи як кількість випадковості в точках вибірки, так і структуру прихованого простору [35].

Висновки до розділу 1

У першому розділі наведена загальна теоретична база щодо архітектури, передумов появи, та принципу роботи відомих на сьогоднішній день моделей генеративного зображення, котрі у свою чергу дедалі швидше та частіше оновлюються, вбираючи в себе більш сучасніші процеси та технології. Також проаналізовано принцип відмінностей двох популярних баз для створення моделей генеративних зображень, а саме Diffusion та GAN моделі.

Підкреслено важливий аспект проблематики навчання моделі ШІ для генерації зображення. Виявлено, що причина важкості навчання моделі генератора та моделі дискримінатора навчаються одночасно. Що у свою чергу значить, що вдосконалення, наприклад, генератора, відбуваються разом з дискримінатором. Для аналізу даних аспектів прийнято рішення проаналізувати етапи навчання ШІ на власному досвіді. Наприкінці першого розділу стисло проаналізовано загальний процес створення GAN моделі власноруч. Побудовано відповідні графіки, що ілюструють загальні етапи навчання моделі. Для цього визначено такі важливі аспекти, як: генерація вибірки з точки функції шляхом генерації випадкових значень. Розроблено Дискримінатор, аби він міг брати зразки з задач. Генератор є етапом, де генерація випадкових точок в латентному просторі починає вчитися.

Для того, аби його навчити проаналізовано модель генератора, яка покликана розробити міру оцінки для генератора. Напрацьований теоретичний фундамент є вкрай важливим аспектом та внеском в подальше проектування ресурсу з впровадженням генеративних зображень за допомогою машинного навчання моделей штучного інтелекту.

РОЗДІЛ 2

ТЕХНОЛОГІЯ СТВОРЕННЯ ГЕНЕРАТИВНИХ ЗОБРАЖЕНЬ, МЕТОДИ ТА ЗАСОБИ ПРОЄКТУВАННЯ ВЕБ-РЕСУРСУ

2.1. Принцип використання AI технології генерації зображення в UI\UX

Визначення ключових «больових» точок

Як користувач, в нього існує ризик зіткнутися з кількома проблемами під час пошуку або створення зображень для веб-сайту та ресурсів дизайну. Деякі з найбільш значущих проблем включають:

Час і вартість. Створення високоякісних зображень вручну може зайняти багато часу та коштувати, особливо якщо відсутні необхідні навички або ресурси. Це може призвести до затримок у проєкті та збільшення витрат [35].

Можливість помилки: Завжди існує ймовірність помилки в процесі виготовлення скетчів і дизайнів вручну. Це може ще більше затримати проєкт і збільшити витрати, а також призвести до розчарування та невдоволення кінцевим результатом.

Ці труднощі можуть розчарувати та перешкодити досягти бажаних результатів проєктування. Однак існують доступні рішення, які можуть допомогти подолати ці перешкоди та спростити пошук і процес створення зображень.

Визначення рішень із больових точок

Створення високоякісних зображень для веб-сайтів і ресурсів дизайну може бути трудомістким і дорогим процесом. Однак існує рішення, яке може допомогти подолати ці проблеми та спростити процес створення зображення.

Одне з рішень полягає в тому, щоб створити платформу генератора зображень зі штучним інтелектом, яка проста у використанні та може створювати високоякісні зображення за менший проміжок часу. Ця платформа використовує штучний інтелект і алгоритми машинного навчання для створення зображень на основі ваших специфікацій, таких як розмір зображення, колірна схема та стиль [36].

Використовуючи платформу для створення зображень AI, з'являється можливість заощадити час і зменшити витрати, пов'язані зі створенням зображень вручну. Крім того, платформа може допомогти уникнути помилок і невідповідностей, які можуть виникнути під час створення зображень вручну без відповідних навичок.

Важливість створення зручної платформи, інтуїтивно зрозумілої та легкої навігації є досить вагомою останнім часом. Таким чином, платформа для створення зображень ШІ повинна бути розроблена з підходом, орієнтованим на користувача, надаючи пріоритет зручності використання, доступності та інклюзивності. Платформа також має бути налаштованою, дозволяючи користувачам налаштовувати налаштування та переваги відповідно до своїх конкретних потреб. Загалом платформа для створення зображень зі штучним інтелектом може бути цінним інструментом для дизайнерів і компаній, які прагнуть оптимізувати процес створення зображень і покращити результати дизайну.

Проблеми створення даної платформи

Завдання при розробці генератора зображень на базі ШІ полягає в тому, щоб переконатися, аби платформа могла створювати високоякісні зображення, які відповідають потребам користувача. Деякі з питань проектування, які необхідно вирішити в процесі, це: чи може користувач переконатися, що платформа може генерувати зображення, які відповідають уподобанням користувача, таким як розмір, колірна схема та стиль? Чи можна гарантувати, що платформа може створювати високоякісні зображення, які не виглядають так, ніби їх згенерувала машина? Чи можна переконатися, що платформа проста у використанні та доступна для користувачів із різним рівнем знань і потребами в доступності?

Альтернативи створення власної платформи

Одним із відомих генераторів зображень зі штучним інтелектом на ринку є MidJourney, інструмент, доступний через всесвітньо відомий застосунок спільноти Discord, який покладається на «prompt».

«Prompt» відіграє вирішальну роль у тому, як MidJourney працює та створює зображення. Він служить вхідними даними або вказівками, які надає користувач, щоб дати директиву моделі штучного інтелекту щодо бажаного вихідного зображення.

Формулюючи підказку, користувачі можуть повідомляти про свої наміри, уподобання чи специфікації, щоб керувати процесом створення зображень MidJourney. Працюючи з MidJourney, користувачі зазвичай надають prompt у формі текстових описів або приблизних ескізів. Ці prompt можуть описувати потрібну сцену, об'єкти, стилі або будь-які конкретні інструкції для розподілу зображення. Наприклад, підказкою може бути «Тихий пляжний захід сонця з пальмами та яскравими кольорами» або «Мінімалістичний дизайн логотипу з чистими лініями та сучасним виглядом». Важливо зауважити, що рівень деталізації та конкретності підказки може вплинути на результат, створений MidJourney. Надання більш конкретних інструкцій може допомогти спрямувати модель штучного інтелекту на створення зображень, які точно відповідають баченню користувача.

Оперативна генерація зображень штучного інтелекту за допомогою MidJourney не тільки полегшує творче дослідження, але й допомагає оптимізувати процес проектування. Дизайнери можуть швидко генерувати різноманітні візуальні концепції на основі різних підказок, що дозволяє їм швидко й ефективно повторювати. Використовуючи потужність підказок, дизайнери можуть використовувати можливості штучного інтелекту MidJourney, щоб розширити свій творчий потенціал і втілити в життя свої дизайнерські ідеї [37].

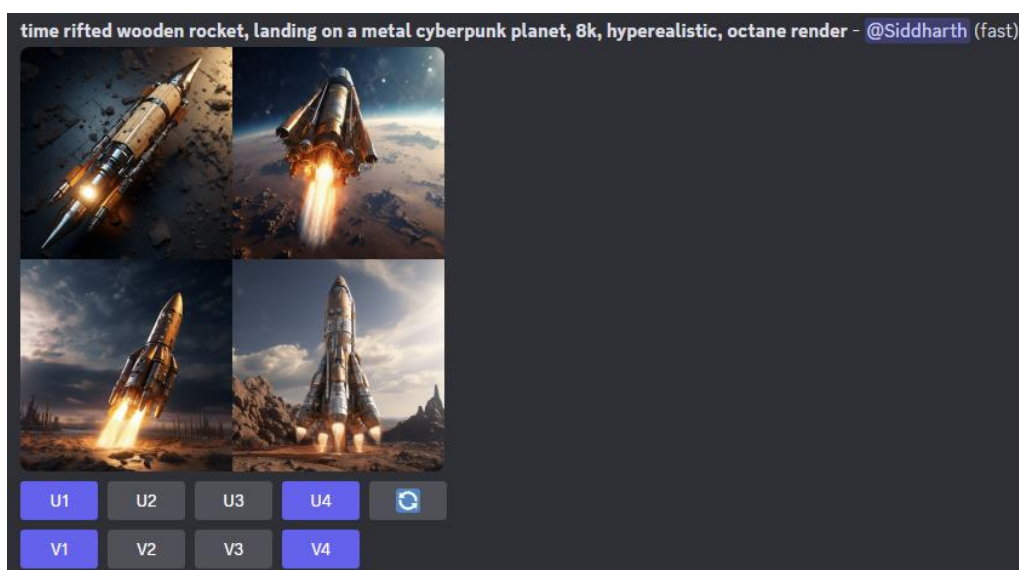


Рис. 2.1. Приклад MidJourney prompt у Discord

2.2. Застосування AI технології генерації зображення як вплив на дизайн

Покращення брифінгу і співпраці з клієнтами

Генератори зображень штучного інтелекту служать ідеальним інструментом для клієнтів, щоб ефективно та візуально передавати свої пропозиції та бачення командам дизайнерів.

Використовуючи генератори зображень штучного інтелекту, клієнти можуть ефективно повідомляти про свої вподобання та вимоги щодо дизайну. Цей візуальний підхід значно покращує спілкування між клієнтами та дизайнерами, зменшуючи ймовірність непорозумінь і гарантуючи, що кінцевий дизайн тісно узгоджується з баченням і цілями клієнта [38].

Прискорення мозкового штурму і етап дослідження

Дослідження в галузі дизайну є фундаментальним аспектом захоплення та визначення суті бренду, продукту чи послуги. Це передбачає вивчення цільової аудиторії, ринкових тенденцій, аналізу конкурентів і розуміння цілей сі. Однак ця фаза дослідження часто може бути тривалим процесом помилок, який іноді не помічається або недооцінюється через брак часу чи ресурсів, що призводить до нерівних результатів.

Для таких компаній-розробників програмного забезпечення, як Dіgox, використання генераторів зображень штучного інтелекту, таких як MidJourney, може значно покращити та прискорити дослідницьку фазу проектування. Ці інструменти є потужним засобом для створення дошок настрою та швидкого пропонування клієнтам дизайну. Використовуючи візуальні елементи, створені штучним інтелектом, дизайнери можуть керувати та вирівнювати очікування клієнтів, демонструючи потенційні напрямки дизайну для логотипів, упаковки, візерунків, візуальних елементів, макетів UX/UI тощо.

Дизайнери можуть використовувати такі генератори зображень зі штучним інтелектом, як MidJourney, щоб досліджувати широкий спектр стилів, колірних палітр і композицій, що дозволяє їм ефективно представляти клієнтам кілька варіантів дизайну. MidJourney має можливість працювати з численними графічними стилями.

Приклад застосування технології для дизайну на практиці

Реальний приклад застосування такої технології на досвіді роботи в сфері графічного дизайну може мати такий вигляд: замовник надає дизайнеру брифінг з задачею на зображення певного набору побажань, які за умови виконання матимуть вигляд тестового чи готового для комерції продукту. Наприклад, є наступна задача — зобразити Голівудського актора Шона Пена в сніжних умовах в якості тесту, який може в подальшому використовуватись як референс для реальних зйомок актора.

Використання ШІ в цьому випадку надає нам дві суттєвих переваги:

- зобразити це за допомоги ШІ значно дешевше та ефективніше, ніж здійснювати підготовчий етап з пошуком ідеї власноруч;
- дизайнер може зобразити скетч та ідею без вимог до майстерних навичок, що спрощує реалізацію задачі та пришвидшує увесь процес.

Для реалізації даної задачі взято продукт MidJourney. Як зазначено вище, для реалізації задачі необхідно мати prompt команду, яка дасть ПЗ директиву для виконання поставленої мети. Перш за все, аби ПЗ розпізнало команду генерації — необхідно прописати **/imagine**.

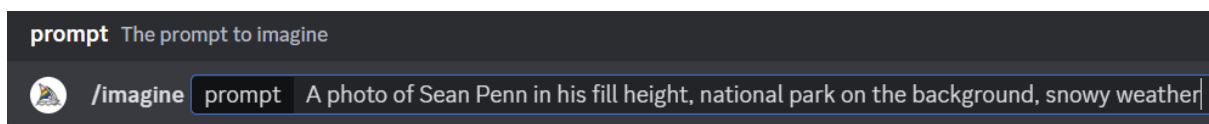


Рис. 2.2. Першочерговий запит (prompt)

В цій команді прописані основні критерії генерації: об'єкт, його зовнішність, певні риси, його оточення з характеристиками та наявності потреби — додаткова інформація, наприклад, погода (це дасть ШІ уяву кольорової гами для сцени та уточнить для ШІ більше критерії, які очікує від нього користувач). Наступним чином, для досягнення реалістичного зображення необхідні додаткові параметри: кут об'єктиву, стиль зображення, плівкове зерно, тип фокусу, модель камери (для цієї задачі замовник може мати певне побажання по візуальній особливості камери, на яку, наприклад, в подальшому зніматимуть реальний продукт) (рис. 24).

Опісля підтвердження генерації зображення з заданими параметрами, ШІ згенерує користувачу чотири зображення з яких користувач зможе обрати те, яке йому до вподоби (рис. 25).



Рис. 2.3. Група згенерованих зображень

Для подальшої роботи, звичайно, одного прикладу буде не достатньо, тому, як правило, генеруються додаткові зображення з можливими правками в запиті та іншому підході до ілюстрації ідеї. По досягненню мети, користувач має наступні можливості: дати ШІ команду згенерувати додаткові варіації до обраного зображення, змінити область зображення, яке йому не до вподоби, а також, найважливіше, виконати апскейл бажаного зображення.

Нижче приведене порівняння якості згенерованого зображення в першочерговій, оригінальній стадії та поліпшеною завдяки вбудованому функціоналу технологій ШІ апскейлінгу (рис. 27).

Для подальшої роботи, із згенерованим зображенням користувач може, наприклад, використати Adobe Photoshop для створення максимально наближеного до реального продукту тестового зразка, наприклад, постеру (рис. 28).



Рис. 2.4. Тестовий постер з використанням згенерованого зображення

Загалом, інструменти штучного інтелекту використовуються в багатьох із цих сфер, наприклад:

- веб-дизайн — створення цільових сторінок;
- motion design — анімаційна графіка, наприклад, мультфільми;
- 3D-моделювання — створення 3D моделей за лічені хвилини;
- промисловий дизайн — допомога дизайнерам у створенні ескізів автомобілів або одягу.

Незважаючи на те, що штучний інтелект є корисним, він не може повністю замінити людей-дизайнерів у жодній із цих сфер, оскільки вони вимагають креативності, гарного смаку та чіткого розуміння вимог бізнесу.

ШІ також не є ефективним у UX/UI та дизайні продуктів, які покладаються набагато менше на фактичні навички проектування, а набагато більше на розуміння бізнес-логіки.

2.2.1. MidJourney у якості генерації візуальних ідей для веб-дизайну

Midjourney має потенціал змінити процес створення інтерфейсів користувача. Завдяки своїй здатності визначати шаблони та генерувати результати, які можуть інформувати ваші дизайнерські рішення, Midjourney може допомогти створити інтерфейси, які будуть не тільки естетично привабливими, але також дуже функціональними та інтуїтивно зрозумілими. Щоб створити скетч інтерфейсу в Midjourney, важливо зауважити кілька моментів — просто покладатися на результати за замовчуванням, які генерує Midjourney, може бути недостатньо для створення унікального веб-сайту чи бренду, що запам'ятовується. Щоб отримати найкращі результати, важливо бути підготовленим із чітким уявленням про те, який проект чи бренд ви хочете створити [40].

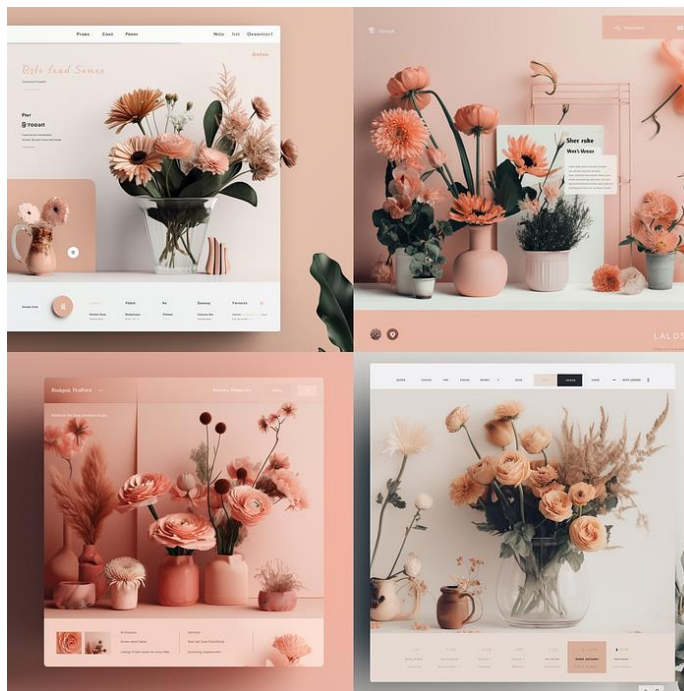


Рис. 2.5. Згенерований в Midjourney тестовий скетч UI інтерфейсу уявного магазину

Одним із найскладніших аспектів використання Midjourney для створення візуальних елементів є його здатність створювати людські елементи. Не можна заперечувати, що результати можуть бути вражаючими, але вони не завжди ідеальні. Для цього прикладу проведений певний тест: як Midjourney створить зображення людських рук, які виконують різні дії, наприклад, загортають букет або збирають квіти, аби поліпшити зв'язок з темою веб-сайту, а саме умовного магазину. Midjourney згенерував кілька результатів, які виглядають багатообіцяючими, але при ближчому розгляді помітні недоліки, які потрібно виправити та/або відтворити в додатковому ПЗ, наприклад використавши Photoshop [41].

Додатково розглянуто спроможність Midjourney генерувати макети веб-сайтів без втручання додаткового ПЗ, аби зрозуміти його максимальні можливості. Для цього тесту створено кілька запитів, які направлені на дослідження ступеню розуміння Midjourney розрізняти стилі веб-сайтів, типографіку та візуальну складову. Перший сайт має бути максимально простим та мінімалістичним, мати заголовок та просту векторну графіку в якості візуальної складової сайту.

Далі наведено відповідний запит: «**web design, a new library, cute illustration simple, flat minimal vector flat, composition, bold lettering --no realistic --ar 3:2**». «>>

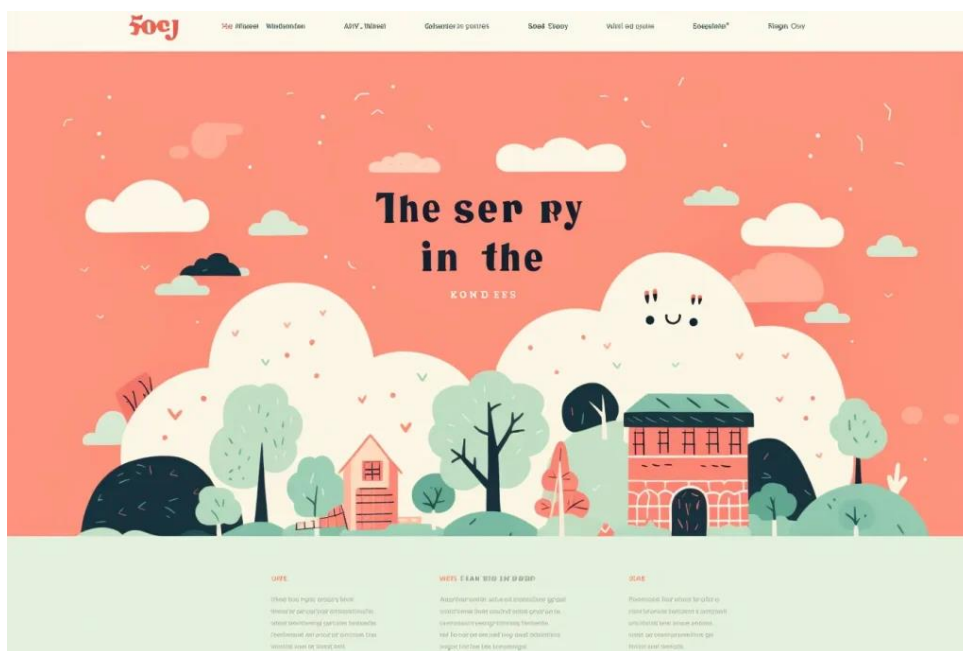


Рис. 2.6. Згенерований макет веб-сайту в Midjourney

Загалом Midjourney розуміє, що структура сайту зазвичай складається з хедеру, де присутній логотип та навігація сайтом, умовний футер з додатковою інформацією про компанію/бренд, ліцензійну та копірайт інформацію та усілякі FAQ блоки, а також розуміє, що body сайту має представляти собою хоча б вид landing page, де є основний заголовок, підзаголовок та background сайту. Недоліком генерації макетів веб-сайту з Midjourney, як й використовуючи аналоги на кшталт Dall-E є, безумовно, нездатність ШІ генерувати відповідний до заданої тематики текст, що так чи інакше змушує користувача використовувати додаткове ПЗ для написання власного тексту, однак ідея використання цього підходу полягає не скільки в готовому рішенні, а скоріше як референс для подальшої будови макету веб-сайту [42].

Наступним чином Midjourney має згенерувати більш серйозний макет сайту з умовами композиційних та типографічних прийомів. Midjourney надано наступний запит: «**web screenshot, the cover of, design & layout, in the style of spatiality, leica cl, frequent use of diagonals, swiss style, matt molloy, bold lettering, monumental forms --ar 16:9**».

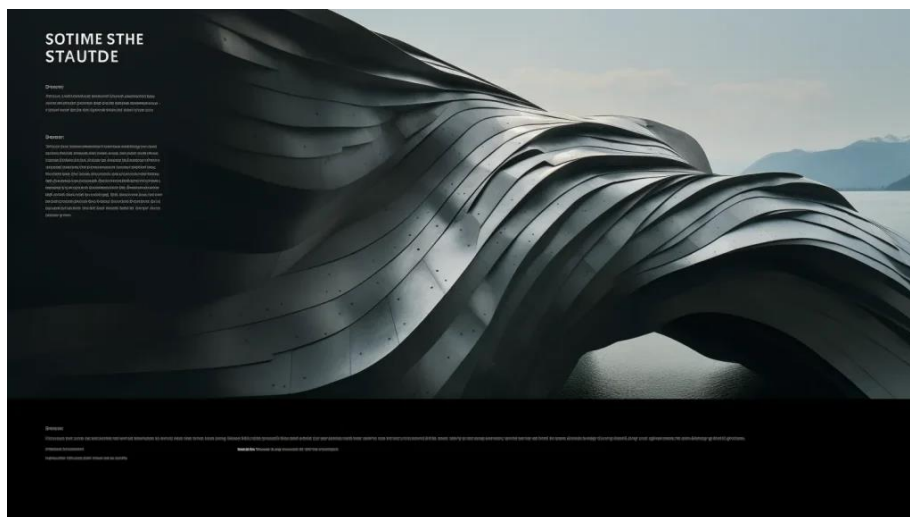


Рис. 2.7. Другий згенерований макет веб-сайту в Midjourney

В цьому прикладі Midjourney демонструє значно цікавіші підходи до реалізації поставленої задачі — він використав композиційні правила та зобразив досить професійний макет веб-сайту.

Ускладнено задачу для Midjourney, адже тепер він має зобразити макет веб-сайту використовуючи лише типографічні прийоми. Запит має наступний вигляд: «**web screenshot, the cover of, design & layout, in the style of spatiality, leica cl, frequent use of diagonals, swiss style, matt molloy, bold lettering, monumental forms --ar 16:9**».



Рис. 2.8. Згенеровані макети типографічного веб-сайту в Midjourney

Останнім тестом є аналіз спроможності Midjourney зобразити абстрактні, градієнтні макети веб-сайту. Запит: «**making good companies look awesome, in the style of website, tinycore, light black and blue, melting pots, rounded, goblincore, minimalist graphic designer --ar 16:9**».



Рис. 2.9. Згенеровані макет веб-сайту в Midjourney

2.3. Вибір ШІ інструментарію для реалізації веб-ресурсу

Художня багатогранність і креативність:

Midjourney виділяється як маяк художньої творчості. Він демонструє вміння вносити унікальний художній штрих у свої результати, особливо коли йому доручено наслідувати стиль певного художника чи твору мистецтва. Зображення, які він генерує, багаті деталями, відображаючи справжні фототехніки щодо світла та кутів. Однак його шлях до ідеального образу може бути важким, вимагаючи ретельного створення підказок для досягнення бажаного результату. Привабливість його мистецьких результатів часто переважає докладені зусилля, що робить його улюбленим для тих, хто шукає поєднання оригінальності та художнього чуття [43].

Навпаки, DALL·E 3 надає пріоритет точності над художньою інтерпретацією. Він вражає, відображає підказку, пропонуючи більш простий, але менш творчий результат. Хоча він намагається включити творчі елементи, посилячись на інших художників або твори мистецтва, він іноді помиляється. Незважаючи на це, візуальна привабливість вихідних даних DALL·E 3 незаперечна, часто створюючи зображення, на які приємно дивитися [44].

Adobe Firefly 2 має тенденцію схилитися до комерційної естетики, створюючи зображення, які здаються готовими для рекламних кампаній. Його підхід до творчості здається більш структурованим, який, хоча й не має мистецької спонтанності як от Midjourney, але забезпечує надійний шлях до створення придатних для використання зображень професійного рівня. У комерційному плані Adobe Firefly 2 вважається найкращим продуктом, оскільки він не лише працює з підказками, але також пропонує розділ налаштувань для швидкого створення підказок і отримання швидших результатів. Функція Style Reference досить цікава, адже вона дозволяє написати підказку, завантажити посилання на стиль, і це створить зображення в завантажених або багатьох доступних стилях. Ця функція, серед іншого, спрощує швидке створення та забезпечує узгодженість стилю, світла, кольору та тону, а також композицію для подібних зображень, що робить її кращим вибором для професіоналів [45].

Можливість ШІ накладати текст на згенероване зображення

Накладення тексту є важливою функцією для брендингу та професійної роботи, і тут DALL·E 3 виділяється серед аналогів. Він демонструє надійну здатність точного розміщення тексту поверх зображень, з чим не можуть зрівнятися ні Midjourney, ні Adobe Firefly 2.



Рис. 2.10. Чіткий та зрозумілий текст згенерований DALL·E 3

Adobe Firefly 2 працює краще, пропонуючи золоту середину з пристойною, хоча й не бездоганною функцією накладення Midjourney, зокрема, бореться з цим аспектом, часто взагалі ігноруючи введення тексту, що може бути суттєвим недоліком для комерційних програм [46].

Простота використання та інтерфейс користувача:

Користувальницькі інтерфейси цих платформ значно впливають на досвід користувача. DALL·E 3, схожий на GPT-4, вражає своєю здатністю генерувати бажані результати незалежно від якості підказки. Його надійність нагадує команду графічних дизайнерів.

Adobe Firefly 2, однак, має інший підхід. Він може похвалитися панеллю налаштувань, яка полегшує оперативне створення та налаштування, тим самим прискорюючи процес створення зображення. Ця функція в поєднанні з опцією посилення на стиль значно спрощує процес створення зображення. Платформа також має високі оцінки щодо інтеграції з іншими продуктами Adobe, що робить її переконливим вибором для професіоналів, які звикли до екосистеми Adobe.

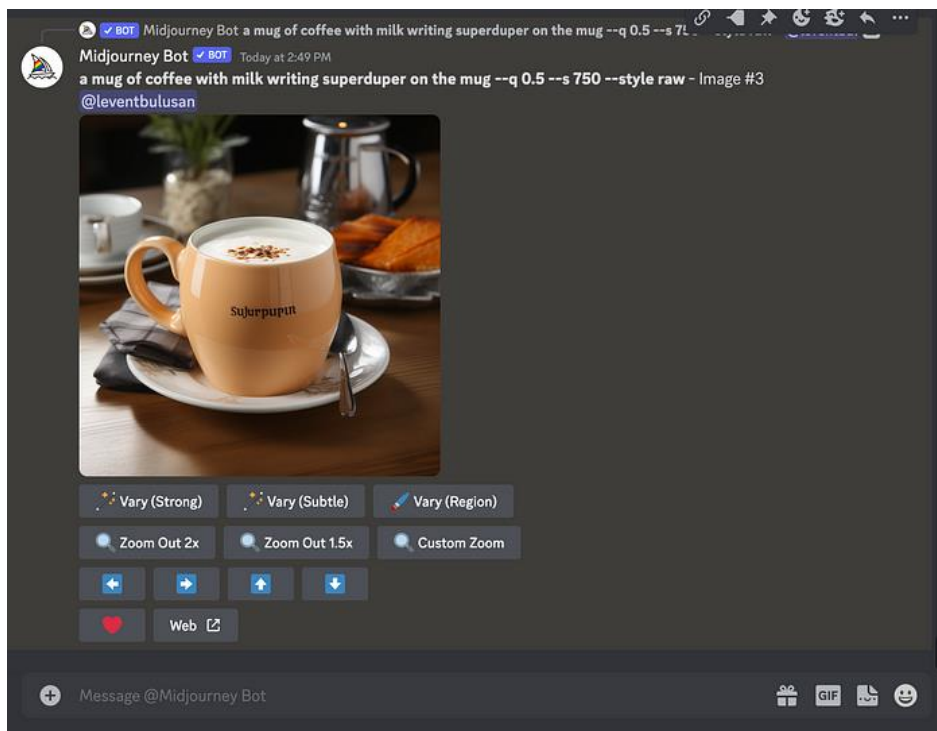


Рис. 2.11. Відсутність власного інтерфейсу Midjourney, приведено текстові команди через Discord

Порівняння зразків запитів

Порівняння зразків між трьома, використовуючи сцену неділі в Парижі в стилі Ван Гога, виявив перевагу DALL·E 3 у бездоганному поєднанні теми та стилю. Його результат був дивовижним поєднанням суті підказки, легко поєднуючи паризьку сцену з культовим стилем Ван Гога. Він блискуче передає суть Парижа та штрихів Ван Гога. Якби штучний інтелект не був відомий, і ви б побачили це зображення, точно подумали, що Ван Гог відвідав Париж, взяв своє полотно та намалював його.

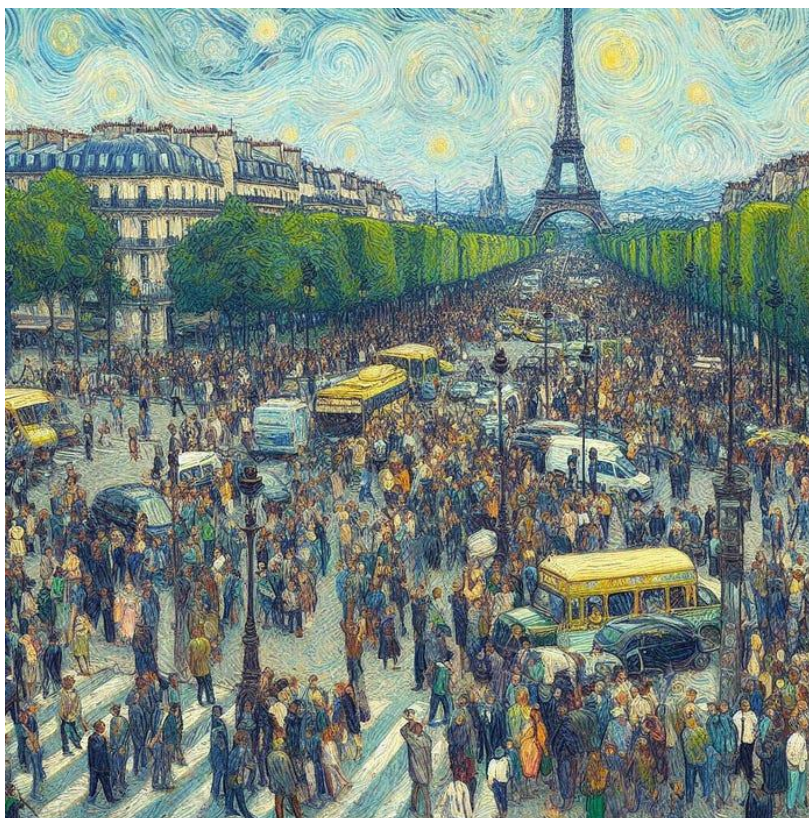


Рис. 2.12. Зразок запиту в DALL·E 3

Adobe Firefly 2 відхилився, змішавши стилі та відвернувшись від теми Ван Гога, тоді як Midjourney, хоч і створив візуально приємне зображення, відхилився від зазначеного художнього стилю, відображаючи його схильність до оригінальності над точністю. Adobe Firefly 2, однак, бере більше паризького аспекту підказки, поєднує його з сумішшю художніх стилів, створюючи заплутаний результат, який, хоча й нагадує Ван Гога, має підказки Моне та кольорову палітру, яку, можливо, можна віднести до стилю французького художника. Зображення не має певного зв'язку з Ван Гогом; правильний лише паризький аспект. Можливо, експерти могли б зробити якісь висновки на основі мазків пензля, але як звичайному ентузіасту мистецтва важко помітити зв'язок.

Midjourney, подібно до Adobe Firefly 2, змінюється самостійно, створюючи чудове зображення Парижа та Ейфелевої вежі, на яке приємно дивитися або навіть повісити на стіну. Зображення, безумовно, приємне візуально, але суть підказки та отримане зображення значно відрізняються.



Рис. 2.13. Зразок запиту в Midjourney

У підсумку, вибір між Midjourney, DALL·E 3 і Adobe Firefly 2 залежить від конкретних потреб і вподобань користувача. Будь то мистецька спонтанність Midjourney, оперативна точність DALL·E 3 чи комерційне використання Adobe Firefly 2, кожна платформа займає унікальне місце в середовищі створення зображень за допомогою ШІ, що швидко розвивається.

2.3.1. Вибір між Dall-E 2 та Dall-3 для інтеграції в веб-ресурс

Після того, як OpenAI представила GPT-4 у червні 2022 року, вони звернули увагу на наступний пункт у своїй дорожній карті: доопрацювання та запуск свого DALL-E 3. Нова модель може конкурувати з Midjourney як у креативності, так і в нюансах. Якщо необхідно зобразити чистий реалізм, об'єктивно перевага за DALL-E 2 над DALL-E 3. Літній чоловік на його зображенні має обличчя людини, яке дійсно може бути реальним: зморшки, лінії сміху, пігментні плями тощо. Справа не в тому, що DALL-E 3 впорався погано. Однак він має тенденцію згладжувати обличчя, що іноді призводить до того, що обличчя виглядають як воскові фігури.



Рис. 2.14. Реалістичний портрет літньої людини крупним планом
(Dall-E 2 ліворуч / Dall-E 3 праворуч)

На наступному тесті помітна еволюція DALL-E 3 порівняно з попередньою версією. Замість того, щоб створити щось узгоджене, DALL-E 2 виглядає так, ніби він зібрав елементи в підказці в одне роз'єднане зображення. З іншого боку, DALL-E 3 створив зображення, яке більше відповідає суті підказки. Він затишний і відповідає всім вимогам без надмірної компенсації.

На тесті з спробою зобразити реалістичний та історичний контекст DALL-E 2 повністю промахнувся. DALL-E 2 створив погане зображення таксофона. Немає елементів, які кричать про «Нью-Йорк» чи щось конкретне про 1980-ті. DALL-E 3 також мав труднощі з цією підказкою. Об'єктивно третя версія впоралась краще, але не достатньо. По-перше, DALL-E 3 згенерував чорно-біле зображення, коли кольорова фотографія була звичним явищем у 1980-х роках, і по-друге, метро над землею — дуже хибне явище.

На тесті з спробою зобразити сюрреалістичний сценарій, пейзаж у DALL-E 2 достатньої якості, але друга версія не виявила спроб створити гібрид людини та лисиці як це зазначено в запиті. DALL-E 3 повністю зрозумів контекст і навіть додав власну родзинку в кінцевому результаті. Наразі це одне з найцікавіших порівнянь, оскільки воно ідеально відображає, наскільки далеко просунулася нова модель.



Рис. 2.15. Гібрид людини та лисиці, що гуляє в сюрреалістичному пейзажі мрії, який поєднує в собі елементи лісу, пустелі та тропічного пляжу (Dall-E 2 ліворуч / Dall-E 3 праворуч)

На тесті з спробою зобразити архітектурний контекст, будинок з DALL-E 2 явно страждав від проблем із рендерингом. Якщо збільшити масштаб, одразу помітно, що він намагався створити балкон з правого боку другого поверху, але він перетворився на дивний гібрид вікна та балкона.



Рис. 2.16. Сучасний двоповерховий, екологічно чистий та стійкий будинок із сонячними панелями, системами збору дощової води та зеленим садом на даху (Dall-E 2 ліворуч / Dall-E 3 праворуч)

Рослини також закривають більшу частину першого поверху, через що важко побачити, як насправді виглядає будинок. Щодо DALL-E 3, то тепер це дійсно схоже на будинок, виняткове зображення сучасного екологічного будинку. Однак версія DALL-E 3 зображає генерації надто мрійно, що зменшує його реалістичність.

На тесті з спробою зобразити 3D рендер DALL-E 2 знову пропустив важливий аспект запиту, створивши ілюстрацію замість 3D-діорами. DALL-E 3 зміг створити зображення діорами, що містить увесь наданий контекст.

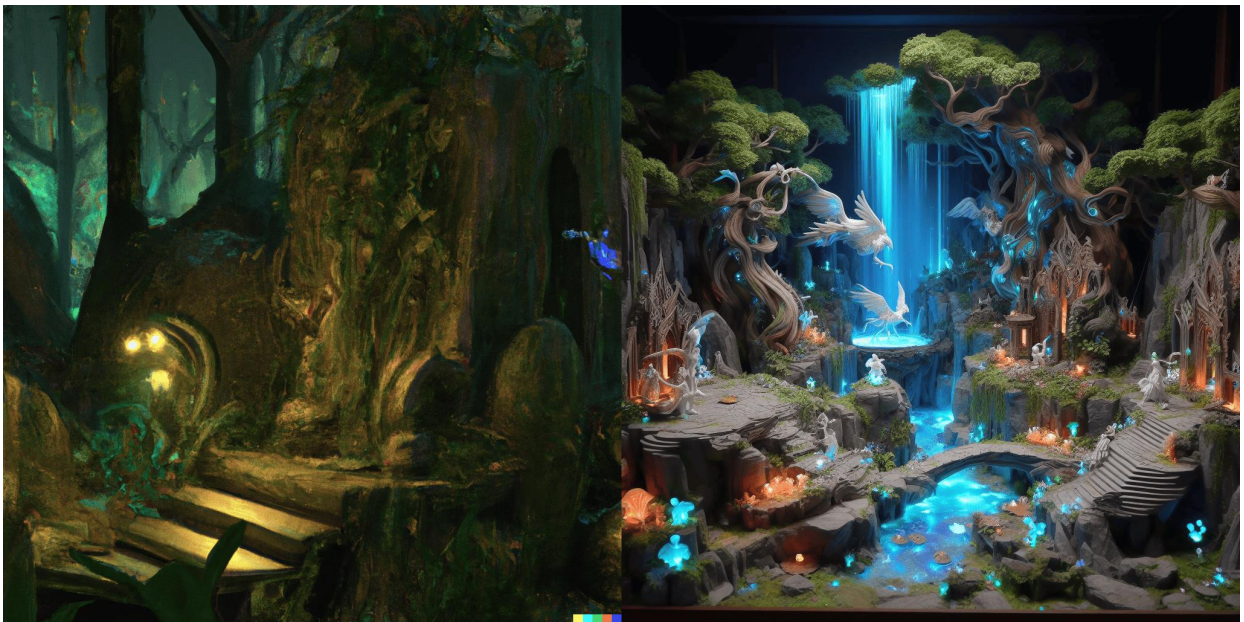


Рис. 2.17. 3D-діорама фантастичного лісу з міфічними істотами, стародавніми руїнами та чарівними біолюмінесцентними рослинами (Dall-E 2 ліворуч / Dall-E 3 праворуч)

На тесті з реалістичним фото крупним планом DALL-E 2 досить добре створює зображення крупним планом. Насправді жодне з двох зображень не є поганим, але об'єктивно перевага DALL-E 3 через контраст і освітлення.

Не зважаючи на переваги DALL-E 3, для інтеграції в проєктований веб-ресурс попередня версія не виглядає значно гіршим рішенням через те, що безпосередня ідея інтеграції даної технології має на меті продемонструвати сам принцип роботи генерації зображення з чим DALL-E 2 здатен впоратись.

2.4. Вибір ПЗ для проєктування веб-ресурсу

Для безпосереднього проєктування веб-ресурсу вибір лежить між двома популярними платформами на сьогоднішній день, а саме Squarespace та WordPress. Однак перш за все слід визначити особливості, переваги та недоліки двох платформ задля прийняття подальших рішень.

Простота використання та крива навчання

WordPress — найкращий вибір серед користувачів, які хочуть створити блог, або власників малого бізнесу, які створюють веб-сайт своїми руками. WordPress також вважається найкращим конструктором веб-сайтів на ринку. Понад 43% усіх веб-сайтів в Інтернеті використовують саме WordPress. Незважаючи на те, що WordPress досить інтуїтивно зрозумілий і простий у використанні, у ньому присутня крива навчання (рис. 50).

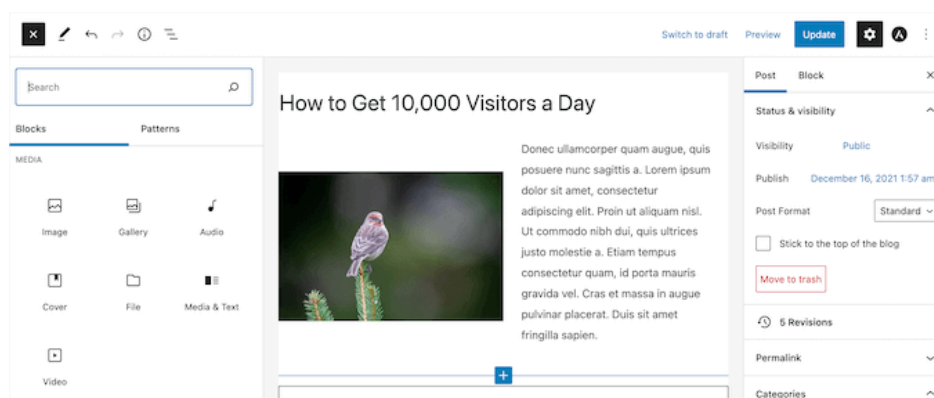


Рис. 2.18. Стандартний редактор вмісту WordPress має простий й зручний у використанні інтерфейс користувача

На відміну від WordPress, Squarespace не має великого вибору. Ця відсутність вибору робить його дуже простим і зручним для користувача. Навіть абсолютні новачки можуть швидко написати вміст і опублікувати його.

Squarespace також використовує редактор блоків, схожий на WordPress. Додавати зображення, відео та аудіофайли простіше, ніж у WordPress. Є багато блоків вмісту, і кожен блок має кілька варіантів керування макетом.

Squarespace дозволяє налаштовувати шаблони та сторінки за допомогою власного візуального редактора. Однак користувачі обмежені параметрами, що доступні налаштуваннями, оскільки це контрольоване середовище.

Дизайн і шаблони

WordPress і Squarespace постачаються з готовими шаблонами та інструментами дизайну, які полегшують вам створення веб-сайту на замовлення. Нижче приведено аналіз пропозицій двох платформ серед можливостей і інструментів для створення добре розроблених веб-сайтів.

WordPress пропонує доступ до тисяч безкоштовних і преміальних тем (шаблонів веб-сайтів). Ці шаблони легко налаштовуються, що дозволяє використовувати логотип й кольори веб-сайту. Багато тем WordPress мають кілька варіантів макета, редактор перетягування, повзунки, фотогалереї та масу інших функцій.

Squarespace пропонує готові шаблони веб-сайтів, охайно організовані за різними категоріями. Є шаблони Squarespace для всіх популярних категорій веб-сайтів із професійним дизайном, який добре виглядає на всіх пристроях. Користувачі можуть легко вибрати шаблон для свого веб-сайту Squarespace і змінити його за допомогою вбудованого налаштувача.

Однак це велика область, де Squarespace серйозно відстає. Кількість попередньо створених шаблонів дуже обмежена, і варіанти дизайну також обмежені набором функцій, доступних у налаштувачі Squarespace. Теми постачаються з обмеженою кількістю варіантів макета з кожним шаблоном. У користувачів все ще можете налаштувати кольори, шрифти, логотипи веб-сайтів та інші речі, але ці шаблони не так налаштовуються, як теми WordPress. Однак це не означає, що користувачі зобов'язані використовувати вбудований набір тем обраний платформою Squarespace, адже користувачі мають змогу створити сайт з нуля підбираючи кожен окремий елемент дизайну веб-сторінки власноруч. Це стосується не тільки кольорової палітри окремих дизайн блоків на сайті, але й, безпосередньо, шрифтові гарнітури й їх накреслення згідно з примхами користувачів [50].

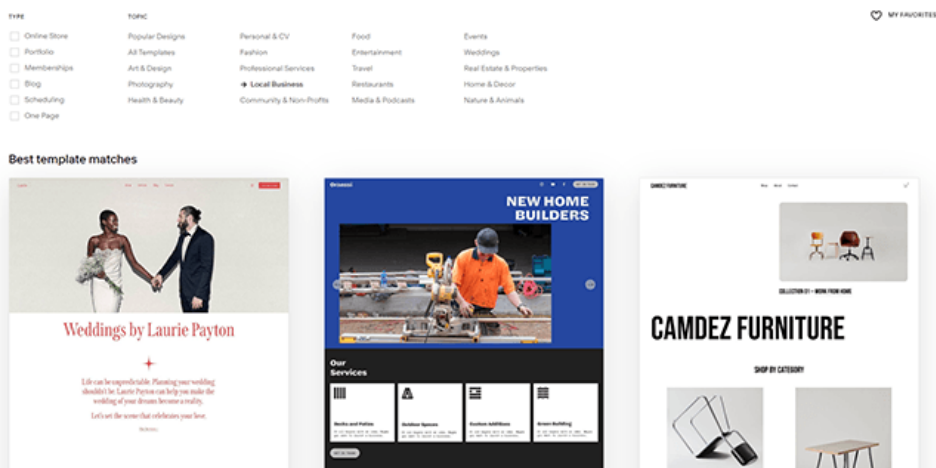


Рис. 2.19. Інтерфейс користувача Squarespace

У підсумку WordPress виглядає привабливіше з точки зору вибору дизайну, параметрів налаштування, функцій створення веб-сайтів і гнучкості. Squarespace пропонує чудові шаблони, але вони обмежені в кількості та не дуже гнучкі.

Розширення та інтеграції

Усі конструктори веб-сайтів мають набір вбудованих функцій. У міру того, як веб-сайт росте, користувачі хочуть додати більше функцій і підключити свій сайт для роботи зі сторонніми інструментами та службами. WordPress має величезну екосистему плагінів. Ці плагіни схожі на програми для вашого сайту WordPress, які ви можете встановити, щоб додати нові функції або змінити типову поведінку вашого сайту. У користувачів є спеціалізовані плагіни для створення конкретних веб-сайтів, наприклад:

- сайти електронної комерції;
- членські спільноти;
- продаж онлайн курсів.

Squarespace поступається WordPress, коли мова йде про розширюваність та інтеграцію. Він пропонує дуже обмежений набір розширень та інтеграцій. Однак, на відміну від WordPress, Squarespace не має потужних API, які розробники могли б використовувати. Доступні інтеграції часто обмежені у функціональності, і користувач не може використовувати їх так вільно на Squarespace, як на WordPress.

Висновки до розділу 2

У другому розділі наведена практична інформація щодо важливості та гнучкості використання 3D моделей генерації зображень в різноманітних галузях дизайн та мультимедійної сфери. Наприклад, згідно завдання кваліфікаційної роботи необхідно розробити веб-ресурс, розробка якого у свою чергу відноситься до галузі UI/UX. Тож враховуючи це проаналізовано принцип використання 3D в галузі UI/UX: виконано пошук можливих проблематик, способів вирішення проблем та перспективи застосування технології 3D в даній галузі. Також наведено практичний досвід з використання даного 3D під час роботи з клієнтами. На власному досвіді описано процес роботи над умовним проектом, що вимагає генерації зображення за допомоги 3D.

Окрім цього детально проаналізовано можливості кожного з даних 3D безпосередньо створивши критерії оцінки для кожної з моделі на ринку. Наостанок опісля пітчингу усіх доступних та цікавих моделей завдяки безпосередньому порівнянню двох генерацій обраного 3D та наведенню ілюстративного аналізу переваг та недоліків — обрано пасуючий для задачі проектування варіант 3D для інтеграції в проєктований веб-ресурс. Обрано та проаналізовано дві популярні платформи для проектування майбутнього веб-ресурсу. В результаті порівняльного аналізу найліпшим та найдоступнішим для виконання бажаних умов обрано веб-платформу розробки сайтів Squarespace.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-РЕСУРСУ З ВПРОВАДЖЕННЯ ГЕНЕРАТИВНИХ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

3.1. Загальні теоретичні відомості по розробці структури веб-ресурсу

Основні складові, що формують структуру веб-ресурсу зазвичай виглядають наступним чином, незалежно від структурної моделі, добре розроблені веб-сайти мають плавний та інтуїтивно зрозумілий потік користувачів.

Дослідження конкуренції — знайти веб-сайти, які мають спільні функції та цільову аудиторію. Поставити себе на місце звичайного користувача та відчувати, як це — шукати певну інформацію. Вивчити їхні структури та звернути увагу на те, що вони роблять правильно, а що неправильно. Таким чином з'являється можливість дізнатися більшість конкурентів у своїй ніші, які використовують ту саму структуру. Наприклад, онлайн-портфоліо зазвичай мають перетинчасті моделі (*webbed models*). Вивчення того, що відвідувачі можуть очікувати від веб-сайту на основі конкуренції, допоможе створити структуру сайту, орієнтовану на користувача.

Визначення структури — записати ключові категорії та підкатегорії, які має включати веб-сайт. Маючи все перед собою, визначаємо, яка модель може працювати найкраще, залежно від кількості сторінок, які знадобляться. Для бізнесу електронної комерції лише з одним продуктом варто спробувати лінійну модель (*linear model*), щоб користувачі зосередилися на основній пропозиції. Якщо веб-сайт портфоліо містить багато категорій і зразків, слід спробувати ієрархічну модель (*hierarchical model*), щоб направляти користувачів до певних розділів.

Побудова блок-схему користувача (опціонально) — потік користувача — це діаграма взаємодії користувача від початку до кінця. Слід вирішити, що необхідно, аби користувач зробив, чи то зробив покупку, чи підписався на інформаційний бюлетень, і створити потік користувачів, щоб визначити, як вони туди потраплять.

Покроковий перегляд їхнього досвіду гарантує, що таким чином звужується можливість пропустити жоден крок під час створення структури веб-сайту.

Використання внутрішніх посилань — після створення структури варто розробити спосіб навігації веб-сайтом для користувача. Разом із панеллю навігації чи іншою структурою додаються корисні посилання на підсторінки, щоб перенаправляти людей до відповідної інформації. Сторінка «About» може посилатися на різні розділи портфоліо, а сторінка продукту може посилатися на схожі продукти. Передбачаючи, куди відвідувачі сайту хочуть перейти, покращується навігація та створюється міцніша мережа посилань на веб-сайті.

Визначивши основні складові, що формують структуру веб-ресурсу, обрано ієрархічну структуру веб-ресурсу. Надалі слід детальніше дослідити особливості створення ієрархічної моделі.

Ієрархічна структура веб-дизайну

Загалом, кожен веб-сайт починається з цільової сторінки, яка часто буде саме домашньою сторінкою. З домашньої сторінки користувач переходить до основних розділів свого веб-сайту. Ці основні розділи веб-сайтів часто називають посиланнями на сайти (рис. 53).

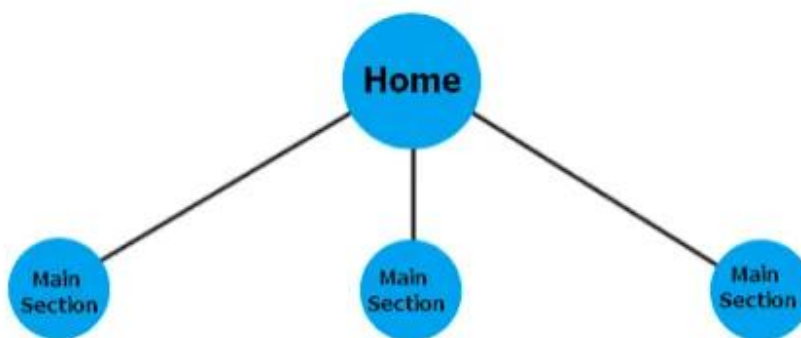


Рис. 3.1. Ієрархічна структура веб-ресурсу

Коли користувач шукає будь-який великий веб-сайт, розробник може побачити, що в результатах пошуку є посилання на інші основні розділи цього веб-сайту.

Наприклад, основними розділами веб-сайту можуть бути, наприклад: «Новини», «Події» або «Блоги». У цих розділах користувачі зможуть досліджувати вміст, що належить до певного типу.

З основних розділів веб-сайту розробник може спроектувати глибші рівні, щоб показати підрозділи. Тут можуть бути розміщені окремі сторінки веб-сайту, або це може бути підрозділ до основного розділу. Наприклад: “Спорт” чи “Політика”, підрозділ у загальному розділі “Новини” або підрозділ для різних регіонів чи тем для розділу подій.

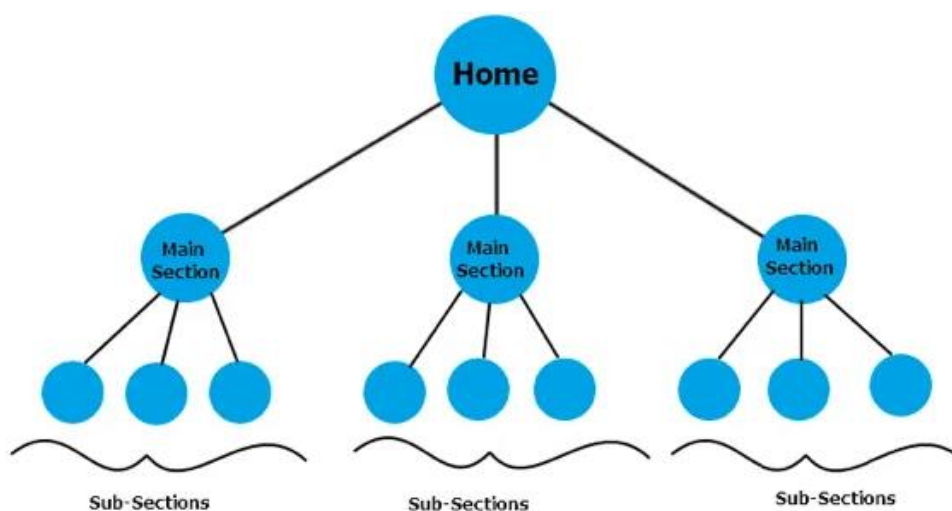


Рис. 3.2. Поглиблена ієрархічна структура веб-ресурсу

Опісля аналізу базових принципів побудови ієрархічної структури слід зануритись у питання планування та практичного впровадження структури веб-ресурсу.

Планування та впровадження структури веб-ресурсу

Щоб реалізувати структуру веб-дизайну, спочатку потрібно скласти план. Цей план слід реалізовувати на ранній стадії процесу проектування. Під час планування структури веб-сайту слід враховувати наступні речі:

- використання принаймні від 2 до 8 основних розділів — все, що містить менше 2 основних розділів, є занадто низьким для веб-сайту, а веб-сайти з більш ніж 8 основними розділами є надто складними;

- усвідомлення контенту з точки зору користувача та уява про завдання, які необхідно, щоб користувачі виконали на веб-сайті, а також про вміст, який може їх зацікавити;

- утримування збалансовану кількість підрозділів, щоб на виході була однакова кількість підрозділів для кожного основного розділу.

- переконання, що сторінки не закопані занадто глибоко в ієрархії веб-сайту, адже користувач заплутається;

- переконання, що вказані основні розділи у верхньому та нижньому колонтитулах веб-сайту. Основні розділи вмісту веб-сайту повинні бути доступні користувачам на будь-якій сторінці та в будь-який час.

Елементи навігації на сайті

Щоб створити хорошу структуру веб-сайту, слід знати концепцію веб-навігації. Веб-навігація полягає в тому, як або які різні елементи на веб-сторінці полегшують спілкування користувачів і перехід від однієї сторінки до іншої.

На рисунку нижче продемонстровано деякі з основних елементів навігації, які використовуються на веб-сайті (рис. 55).

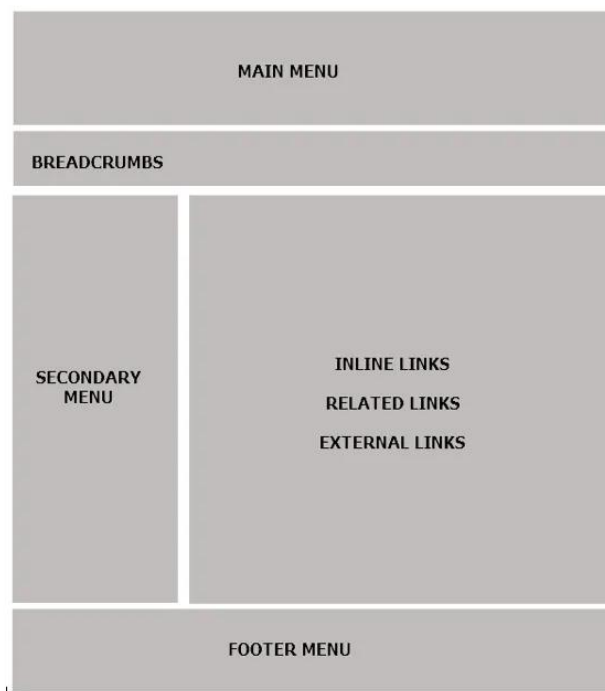


Рис. 3.3. Приклад елементів навігації на макеті веб-ресурсу

Кожна з навігаційних областей використовується для виконання унікальних навігаційних потреб відвідувачів веб-сайту.

Головне меню

Головне меню є найважливішим елементом навігації, і воно в основному розташоване в заголовку веб-сайту.

Додаткове меню

Як випливає з назви, це меню є вторинним, що у свою чергу означає, що воно призначене для підрозділу або основного розділу. Зазвичай він розташований на лівій навігаційній панелі, яку часто називають бічною панеллю.

“Bread Crumbs”

“Bread Crumbs” є дуже важливим елементом навігації, який часто ігнорується. “Bread Crumbs” пояснюють користувачеві, де вони знаходяться на веб-сайті прямо зараз, коли вони потрапляють на певну веб-сторінку вашого веб-сайту з пошукових систем, таких як Google, або інших зовнішніх веб-сайтів. “Bread Crumbs” є надзвичайно важливим елементом веб-навігації для будь-якого веб-сайту, оскільки вони пояснюють користувачам розділ веб-сайту, на який вони потрапили.

Вбудовані посилання, пов’язані посилання та зовнішні посилання

Це різні посилання, які ви можете мати в області вмісту вашого веб-сайту. Ці посилання заохочують більше взаємодії з користувачами та роблять досвід перегляду для аудиторії вашого веб-сайту ще кращим.

Меню нижнього колонтитула

Посилання нижнього колонтитула розташовані у нижньому колонтитулі сайту. Вони являють собою ще один важливий навігаційний елемент. Це допоможе користувачам знайти посилання на головні сторінки вашого веб-сайту, навіть якщо вони прокручують сторінку донизу. Меню нижнього колонтитула також містить деякі рідко використовувані посилання, як-от ваші посилання, політика конфіденційності, положення та умови, карта сайту тощо; це не перші сторінки для відвідування аудиторією веб-сайту. Тому розробник використовує навігацію нижнього колонтитула, щоб надати ці посилання.

3.1.1. Підготовка загального дизайну проєктного веб-ресурсу

Першочергові кроки створення сайту в Squarespace

Під час створення нового сайту Squarespace полегшує процес налаштування сайту користувачеві, наприклад, пропонуючи вибрати тематику веб-ресурсу (рис. 56):

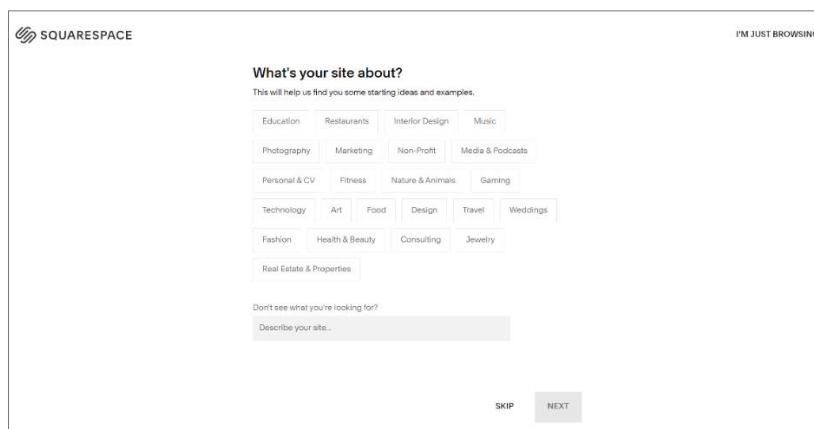


Рис. 3.4. Інтерфейс опису тематики сайту

Таким чином, Squarespace визначає, який з вже готових макетів можна запропонувати користувачеві як подальший фундамент для розробки власного сайту. Для власної розробки обрано тематику: “Technology” та більш звужено пошук макету надавши Squarespace опис майбутнього сайту (рис. 57) [50].

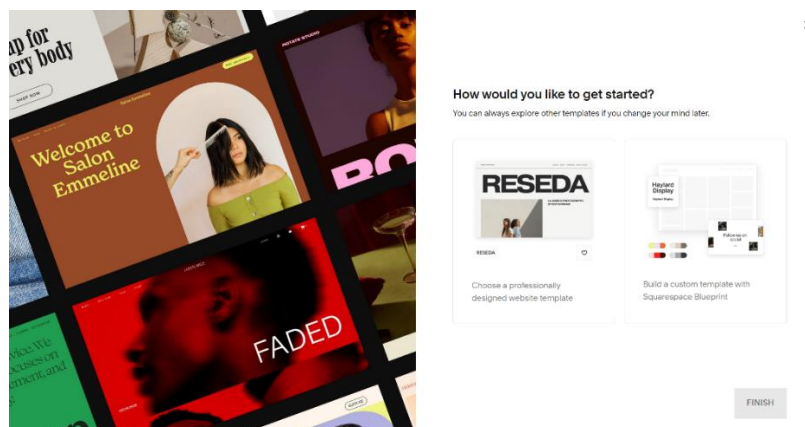


Рис. 3.5. Процес створення веб-ресурсу

З усіх запропонованих варіантів обрано найбільш стриманіший та гнучкий для подальших змін варіант макету (рис. 58). Підтвердивши вибір, Squarespace перенаправляє користувача у власний конструктор, де користувач безпосередньо й працює з макетом (рис. 59):

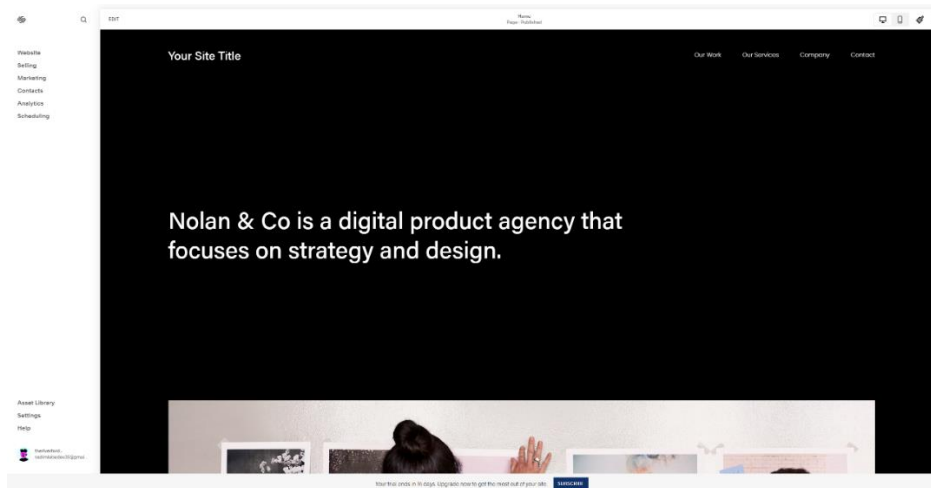


Рис. 3.6. Робоча зона користувача

Задача розробки веб-ресурсу полягає в простому та інтуїтивному дизайні, який підкреслює тему платформи для генерації зображень завдяки штучному інтелекту, поєднує цей функціонал з вбудованим чат-ботом та користувацьким гайдом з користування даними функціями. Для цього необхідно побудувати загальний план структури веб-ресурсу (рис. 60).



Рис. 3.7. Загальна структура сайту

Аби логотип доповнював концепт сайту, розроблено повноцінний вигаданий продукт платформи генерації зображень з назвою **A.A.W.U(M)**, що у свою чергу означає **Autonomous Adaptive Workflow Utilization Module** — Автономний Адаптивний Модуль Управління Робочими Процесами (рис. 61).



Рис. 3.8. Логотип веб-ресурсу

Розроблено хедер веб-ресурсу. Впроваджено створені сторінки веб-ресурсу, додано логотип, посилання на умовні соц.мережі, а також впроваджено функціонал персонального кабінету користувачів. Головна сторінка веб-ресурсу має одразу привертати увагу користувача, тому контрастний логотип у поєднанні з інтерактивним фоном композиційно розташований на макеті (рис. 63).



Рис. 3.9. Логотип як основний композиційний важіль головної сторінки

Додано кнопки кол-ту-екшн та навігації. Нижче користувач скролом миші натрапить на приклад використання генератора зображення (рис. 64).

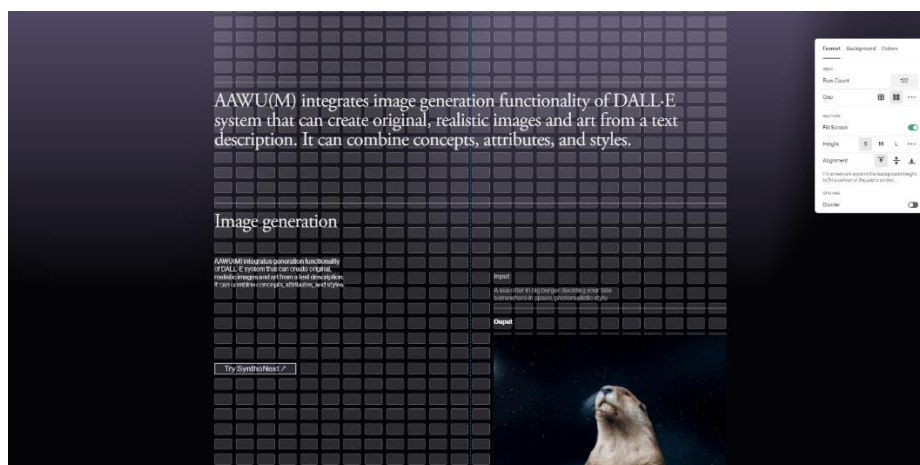


Рис. 3.10. Продовження головної сторінки

Наприкінці користувач зможе наочно оцінити функціонал генераторів зображень. Для цього додано відео-програвач з URL посиланням на відеоресурс (рис. 65).

3.2. Впровадження додаткових фрейворків у функціонал веб-ресурсу

Для впровадження функціоналу генерації зображення за допомогою ШІ створено власний код інтеграції портативної версії масиву функцій платформи Dall-E від розробника Open AI, аби мати можливість використовувати власний фреймворк Server-Side з можливістю відправки запиту Client-Side і навпаки, таким чином створюючи автономний генератор зображення. Скрипт коду з використанням API бібліотеки InteraxAI виглядає наступним чином:

```
<script id="interaxai-widget-config">
  window.INTERAX_AI = {
    widgetID: "dall-e",
    ownerID: "653933478f2c193f6e1bfd09",
  };
</script>
```

```
</script>
```

```
<script id="interaxai-widget-script" defer="defer"
```

```
src="https://cdn.jsdelivr.net/gh/interaxai/cdn@v1.16.0/dist/bundle.js"></script>
```

Бібліотека арі InteraxAI надасть можливість використати базовий фреймворк платформи Dall-E у виді багатофункціонального віджету який інтегрується в воркспейс платформи Squarespace:

```
<interaxai-input id="input1"></interaxai-input>
```

```
<interaxai-select id="select1"></interaxai-select>
```

```
<interaxai-submit id="submit1"></interaxai-submit>
```

```
<interaxai-result id="result1"></interaxai-result>
```

Результатом додавання цього функціоналу є компактний й зручний віджет з полем для вводу інформації та отриманням результату безпосередньо на сторінці веб-ресурсу (рис. 65)

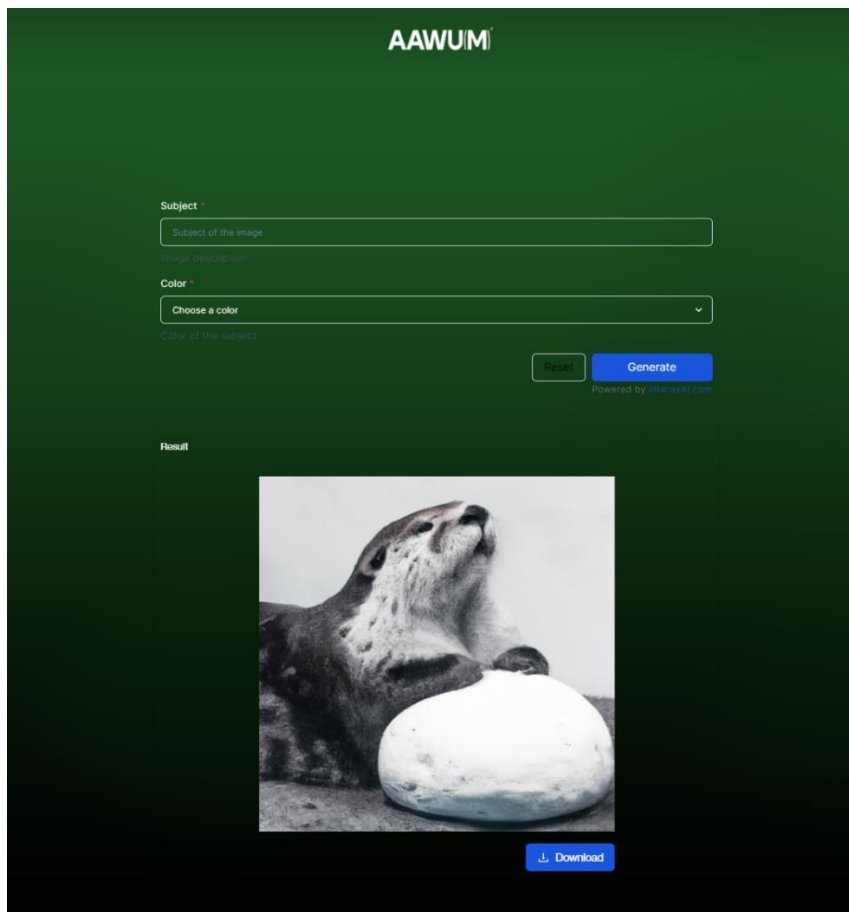


Рис. 3.11. Інтерфейс генератора зображення

Також додано функціонал чат-боту для генерації тексту як окрему сторінку веб-ресурсу. За аналогією з Dall-E впроваджено функціонал платформи Chat-GPT 3.5, який також має вбудований функціонал новітньої версії Dall-E 3.0, однак через компактний інтерфейс та обмежену гнучкість фреймоврку, можливість генерувати зображення через інтерфейс комунікації з Chat-GPT відсутній.

Нижче приведений код інтеграції:

```
<script id="interaxai-widgect-config">
  window.INTERAX_AI = {
    widgectID: "chatbot",
    ownerID: "653933478f2c193f6e1bfd09",
  };
</script>
<script id="interaxai-widgect-script" defer="defer"
src="https://cdn.jsdelivrivr.net/gh/interaxai/cdn@v1.16.0/dist/bundle.js"> </script>
<interaxai-result id="result1"> </interaxai-result>
<interaxai-input id="input1"> </interaxai-input>
<interaxai-submit id="submit1"> </interaxai-submit>
```

Таким чином виглядає інтерфейс Chat-GPT на сторінці веб-ресурсу (рис. 66):

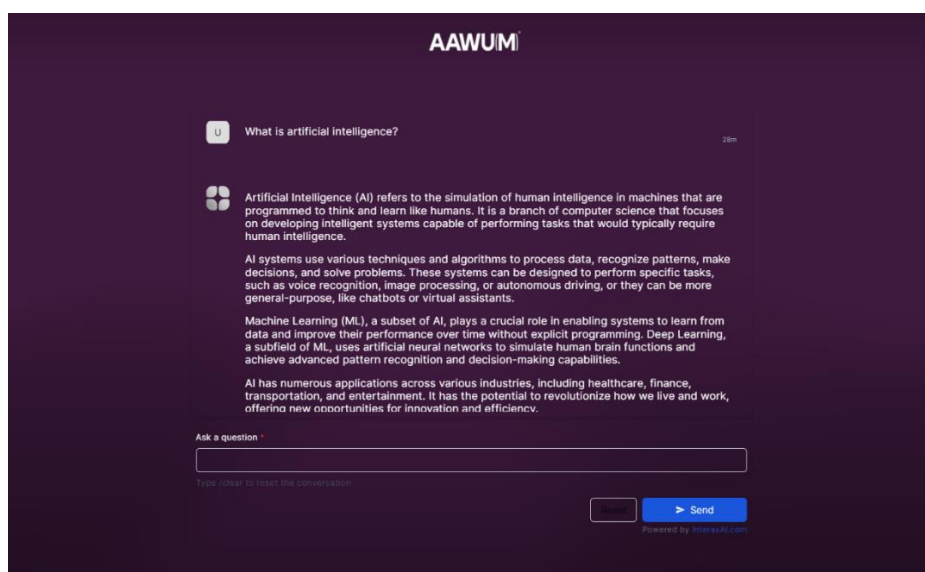


Рис. 3.12. Інтерфейс текстового генератора

В якості допомоги користувачам на сторінці “Guide” додано спеціальну версію Chat-GPT, яка спроектована під навчальні задачі, так званий Educational Chat-Bot. Таким чином користувач зможе отримати більше корисної інформації щодо команд для генерації зображення та різноманітні поради.

3.3. Фіналізування верстки проєктного веб-ресурсу

Сторінка генерації зображення

Після впровадження багатофункціонального віджету на базі Dall-E слід розробити дизайн-макет сторінки. Додано бібліотеку згенерованих зображень користувачем (рис. 67).

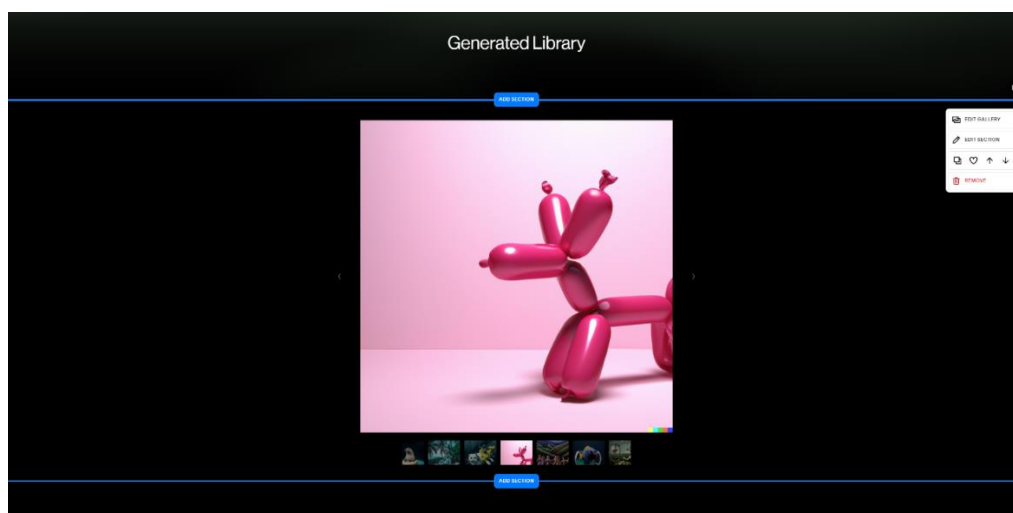


Рис. 3.13. Бібліотека зображень

Також налаштовано процес їх завантажень на веб-ресурс (рис. 68).

Загалом, остаточний вигляд сторінки з генерацією зображень має наступний вигляд (рис. 69). Внесені зміни до футера сайту додавши туди інтерактивне анімаційне зображення (рис. 70). Зверстано сторінку веб-ресурсу, що націлена на розширення функціоналу генерації зображення доповнюючи його можливості. Сторінка “Text Generation” створена, аби користувачі могли отримати від ШІ креативний та цікавий prompt, який потім можна використати для візуалізації.

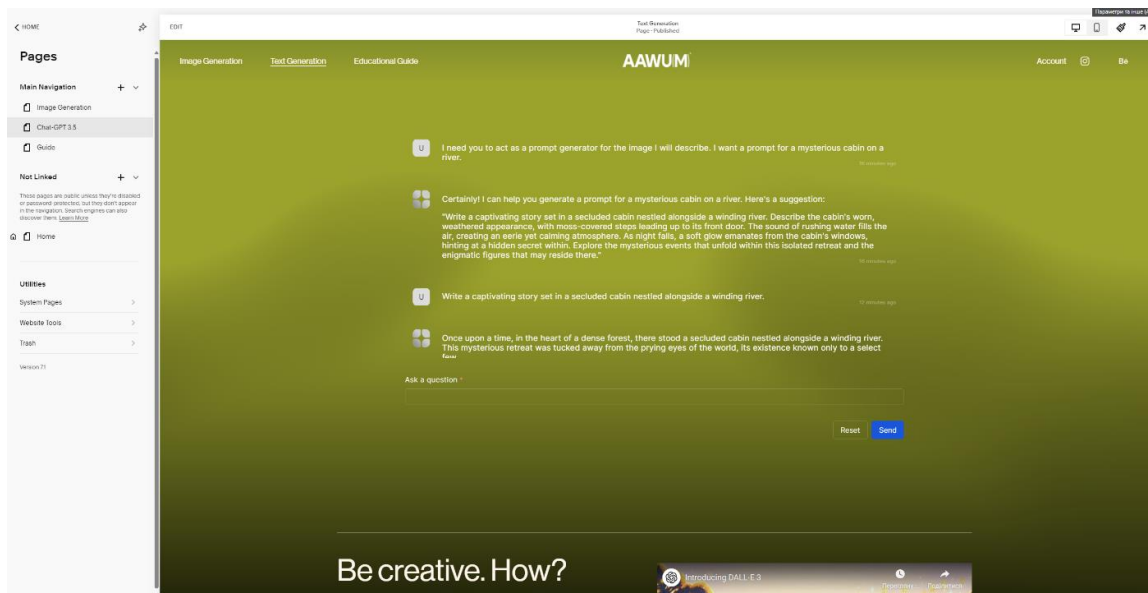


Рис. 3.14. Сторінка генерації тексту

На ній представлено поле для діалогу з ШІ у виді чату, а також короткий гайд, що націлений на допомогу користувачам швидко опанувати можливості взаємодії з даним віджетом. Також додано короткий відео-супровід, який наочно демонструє взаємодію користувача з текстовим генератором. Остання сторінка веб-ресурсу виконана у якості гайд-туторіалу для користувачів. Вона також має вузько-направлену інтеграцію чат-боту.

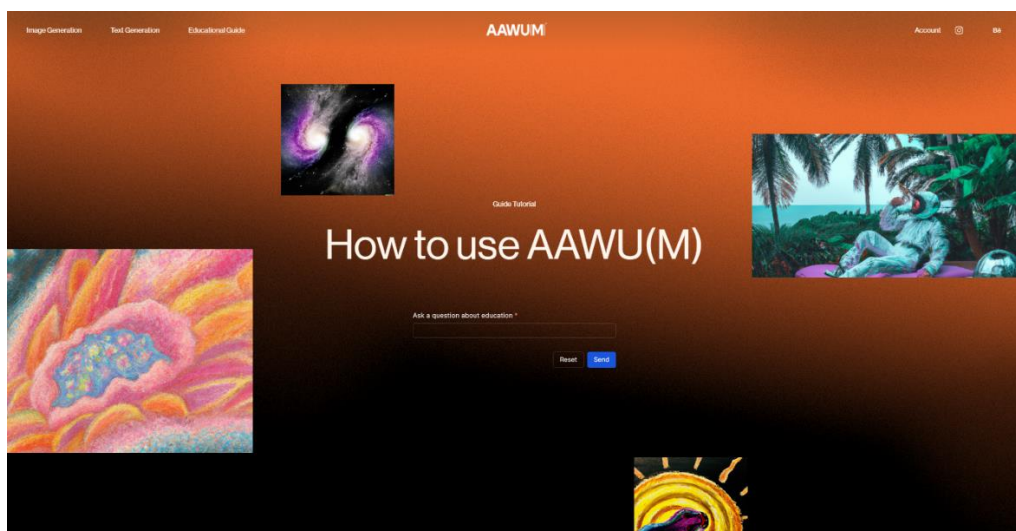


Рис. 3.15. Сторінка Educational Guide

Наостанок для поліпшення процесу навчання для користувачів зроблено короткі відео-нарізки, що демонструють процес взаємодії з моделлю генерації зображення, а також з моделлю текстового генератора (рис. 73).

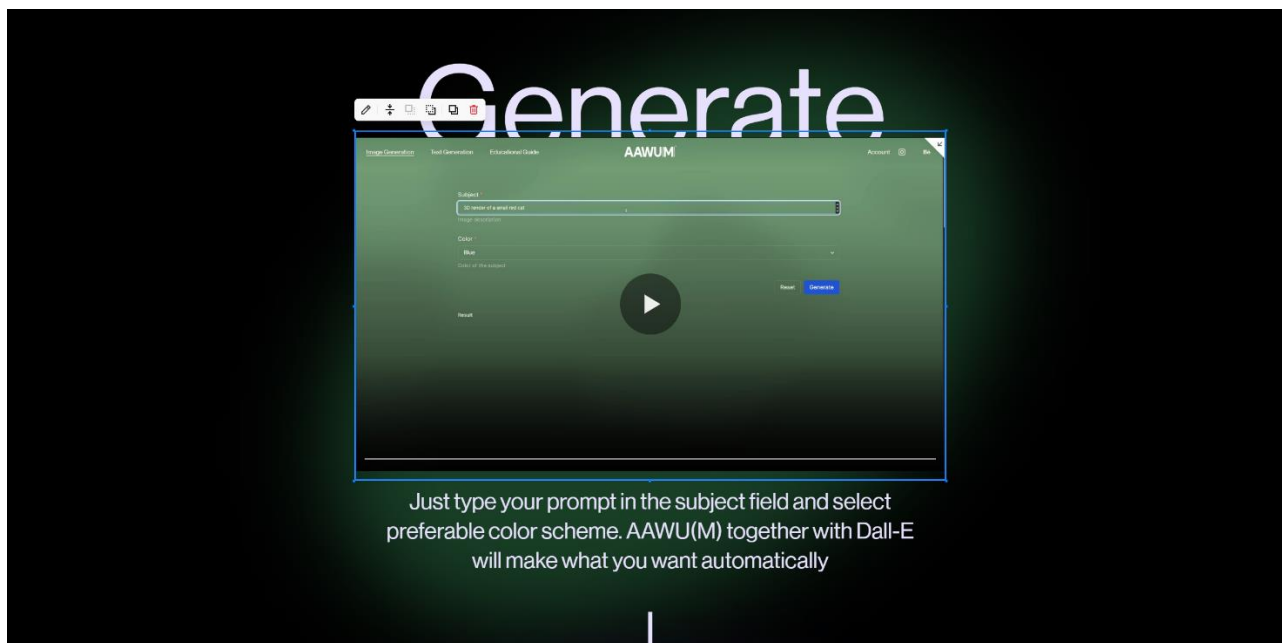


Рис. 3.16. Створений для веб-ресурсу відео-матеріал

Таким чином створено все необхідне для функціонування веб-ресурсу, що надає можливість створити зображення завдяки ШІ.

Висновки до розділу 3

У цьому розділі детально розглянуто поступовий процес створення веб-ресурсу: від інтеграції функціоналу моделей штучного інтелекту на базі зовнішніх, кастомних бібліотек API до графічного оформлення веб-ресурсу. Базуючись на цьому досвіді веб-ресурс отримав кілька окремих вузько-направлених моделей: зовнішній фреймворк Dall-E відповідає за безпосередню генерацію зображення завдяки вводу бажаного запиту в поле для тексту; текстовий генератор ChatGPT-3.5, який обмежений виконувати конкретну поставлену задачу: генерувати для користувача промпт ідей для генерації зображення в Dall-E.

А також додано більш вузько-направлену відокремлену специфіку ChatGPT — допомогти користувачу знайти відповідь на питання завдяки Educational боту. Після налаштування працездатності фреймворків та зв'язку з зовнішніми клієнтами послуг для роботи даних моделей автономно, створено розділове розгалуження, що у свою чергу й спонукало до впровадження різного функціоналу від моделей штучного інтелекту. Продемонстровано загальний процес додавання графічних елементів на сторінки веб-ресурсу завдяки зручному інтерфейсу Squarespace.

Головна сторінка веб-ресурсу виконана в стилі landing page, щоб зацікавити потенційного користувача своїм привабливим видом та текстовим наповненням, аби користувач міг ознайомитись з загальним наповнення сайту. Сторінка “Image Generation” безпосередньо має саму модель генерації зображення, тож до моделі додано функціонал бібліотеки згенерованих зображень. Сторінка “Text-Generation” виконана у подібному стилі, де присутнє поле вводу інформації користувачем, а також короткий відео-гайд, що направлений ознайомити користувача з взаємодією.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є безпосередньо задачі, що поставлені у вступній частині. Аналізування ймовірних теоретичних засад при дослідженні теми: у підсумку виявлено ймовірні теоретичні засади під час роботи з Diffusion моделями генерації зображень, визначено архітектуру та структуру дифузійних моделей, оглянуто механізм синтезу на базі «перетворення тексту в зображення» та синтез «зображення в зображення». Розглянуто модель LDM, що у порівнянні з іншими методами вона не тільки ефективні для пам'яті, але й створює різноманітні, дуже деталізовані зображення, які зберігають семантичну структуру даних.

Також розглянуто такі програми, як Neural Style Transfer (NST): програму, що поєднує вміст одного зображення зі стилем іншого зображення. Виявлено яким чином Diffusion моделі стають найсучаснішими результатами синтезу зображення та інших даних. Крім того, їх формулювання дозволяє створювати керуваний механізм для керування процесом генерації зображення без перенавчання. Не менш важливим стало дослідження синтезу зображень високої роздільної здатності з Latent Diffusion моделями та архітектури GAN у якості перших кроків до становлення Diffusion моделі. Наприкінці розділу надано короткий огляд безпосереднього процесу створення GAN моделі власноруч. У підсумку основних моментів, які виявлено під час ознайомлення з теоретичною складовою: дифузійні моделі працюють шляхом поступового додавання гаусового шуму через серію кроків T до вихідного зображення, процес, відомий як дифузія; для вибірки нових даних ми апроксимуємо процес зворотної дифузії за допомогою нейронної мережі; навчання моделі базується на максимізації нижньої межі доказів (ELBO); моделі дифузії можна обумовити на мітках зображень або вбудованому тексті, щоб «керувати» процесом дифузії. Каскадна та латентна дифузія — це два підходи до масштабування моделей до високої роздільної здатності. Моделі каскадної дифузії — це моделі послідовної дифузії, які генерують зображення зі зростаючою роздільною здатністю.

Моделі латентної дифузії (наприклад, стабільна дифузія) застосовують процес дифузії на меншому латентному просторі для ефективності обчислень, використовуючи варіаційний автокодер для підвищення та зменшення дискретизації. Моделі на основі балів також застосовують послідовність шумових збурень до вихідного зображення. Але вони тренуються за допомогою підбору балів і динаміки Ланжевена. Тим не менш, вони досягають схожої мети. Процес дифузії можна сформулювати як СДУ. Вирішення зворотного SDE дозволяє генерувати нові зразки.

Проаналізовано підхід Latent Space, що починається з аналізу навчених моделей дифузії в піксельному просторі. Як і у випадку з будь-якою моделлю, заснованою на правдоподібності, виявлено яким чином розділити його на етапи: перший — етап, що відповідає за перцептивне стиснення. Він видаляє високочастотні деталі, але все ще вивчає невеликі семантичні варіації. На другому етапі фактична генеративна модель вивчає семантичну та концептуальну композицію даних (семантичне стиснення).

Макетування проектного веб-ресурсу завдяки теоретичним основам: це завдання в повному обсязі розкрито в другому розділі, де наведено загальну розробку дизайн концепту веб-ресурсу (визначено етапи створення дизайн концепту).

Другий розділ також увібрав у собі такі завдання, як аналіз існуючих моделей генерації зображення, проаналізовано сильні та слабкі сторони кожну з конкурентних платформ. У ньому наведена інформація щодо важливості та гнучкості використання ШІ моделей генерації зображень в галузях дизайну та мультимедійної сфери. Виконано пошук можливих проблематик під час застосування ШІ в галузі UI/UX, способів вирішення проблем та перспективи застосування технології ШІ в даній галузі. Також наведено практичний досвід з використання даного ШІ під час роботи з клієнтами. Описано процес роботи над власним проектом, який вимагав генерації зображення за допомогою моделі генерації зображення. Проаналізовано можливості кожного з розглянутих моделей ШІ за критеріями оцінки. Обрано пасуючий для задачі проектування варіант ШІ для інтеграції в проєктований веб-ресурс.

Хоча існує багато дискусій щодо DALL-E 2 та його важливості як для Deep Learning, так і для світу в цілому, варто звернути увагу на три ключові висновки з розробки DALL-E 2: по-перше, DALL-E 2 демонструє потужність дифузійних

моделей у глибокому навчанні, причому як попередні, так і підмоделі створення зображень у DALL-E 2 базуються на дифузії. Незважаючи на те, що дифузійні моделі набули популярності лише за останні кілька років, вони вже довели свою цінність, і ті, хто налаштований на дослідження глибокого навчання, повинні очікувати, що в майбутньому вони побачать більше; другий момент полягає в тому, щоб підкреслити потребу та силу використання природної мови як засобу навчання найсучасніших моделей глибокого навчання. Цей момент не походить від DALL-E 2 (зокрема, CLIP продемонстрував це раніше), але все ж важливо розуміти, що потужність DALL-E 2 в кінцевому підсумку впливає з абсолютно величезної кількості парних даних природної мови/зображень, які доступний в Інтернеті. Використання таких даних не тільки усуває вузьке місце розробки, пов'язане з трудомістким і кропітким процесом ручного маркування наборів даних; але шумна, непідготовлена природа таких даних краще відображає дані реального світу, до яких моделі глибокого навчання повинні бути стійкими. Обґрунтовано вибір програмного забезпечення, що застосовано для проектування.

Третій розділ в повному обсязі описано створення концепту та остаточного дизайн макет веб-ресурсу. Надано детальні кроки досягнення практичної реалізації веб-ресурсу завдяки визначеним в другому розділі програмним засобам та обраній моделі генерації зображення. У цьому розділі розглянуто процес створення веб-ресурсу: проілюстровано інтеграцію функціоналу моделей штучного інтелекту та графічне оформлення. Веб-ресурс у підсумку отримав кілька моделей: зовнішній фреймворк Dall-E; текстовий генератор ChatGPT-3.5, а також додано Educational бот, що направлений на допомогу користувачу знайти відповідь на питання.

Головна сторінка веб-ресурсу направлена на зацікавлення потенційного користувача, аби користувач міг ознайомитись з загальним наповнення сайту. Сторінка “Image Generation” має функціонал генерації зображення. Сторінка “Text-Generation” виконана у подібному стилі, де присутнє поле вводу інформації користувачем, а також короткий відео-гайд, що направлений ознайомити користувача з взаємодією.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ І ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wäldchen J, Mäder P. Machine learning for image-based species identification. *Methods Ecol Evol.* 2018; 9:2216–222.
2. Вступ до моделей дифузії для машинного навчання [Електронний ресурс]. – Режим доступу: <https://encord.com/blog/diffusion-models/> (дата останнього звернення: 20.11.2023). – Назва з екрана.
3. Chakrabarty T. I Spy a Metaphor: Large Language Models and Diffusion Models Co-Create Visual Metaphors / Arkadiy S., Olivia W., Artemis P., Yue Y., Marianna A., Smaranda M. 2023; 2305.14724.
4. Saharia C., Chan W. Palette: Image-to-Image Diffusion Models / Huiwen C., Chris A. Lee, Jonathan H., Tim S., David J. Fleet, Mohammad N. 2021; 2111.05826.
5. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017*; pp. 2223–2232.
6. Пояснення щодо створення зображень ШІ: методи, застосування та обмеження [Електронний ресурс]. – Режим доступу: <https://www.altexsoft.com/blog/ai-image-generation/> (дата останнього звернення: 18.11.2023). – Назва з екрана.
7. Aditya R., Prafulla D., Alex N., Casey C., Mark C. Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022; 2204.06125.
8. Xinyang Z., Wentian Z., Xin L., Jeff C. Text2Layer: Layered Image Generation using Latent Diffusion Model. 2023; 2307.09781.
9. Shikun S., Longhui W., Junliang X., Jia J., Qi T. SDDM: Score-Decomposed Diffusion Models on Manifolds for Unpaired Image-to-Image Translation. 2023.
10. Репозиторій колекції ресурсів і документів про дифузійні моделі [Електронний ресурс]. – Режим доступу: <https://github.com/diff-usion/Awesome-Diffusion-Models#introductory-posts> (дата останнього звернення: 19.11.2023). – Назва з екрана.

11. Brownlee J. Generative Adversarial Networks with Python Deep Learning Generative Models for Image Synthesis and Image Translation. 2019. №. 1. P. 35-96.
12. Kevin F., Lisa B. Soros, Olaf W. Clipdraw: Exploring text-to-drawing synthesis through language and image encoders. 2021; 2106.14843.
13. Ali R., Aaron van den Oord, Oriol V. Generating diverse high-fidelity images with VQ-VAE-2. In NeurIPS. 2019. P. 14837–14847.
14. Tero K., Samuli L., Timo A. A style-based generator architecture for generative adversarial networks. In IEEE Conf. Comput. Vis. Pattern Recog. 2019. P. 4401–4410.
15. Chitwan S, Jonathan H, William C, Tim S, David J. Fleet, Mohammad N. Image super-resolution via iterative refinement. CoRR, 2021; 2104.07636.
16. Gaurav P, Dacheng L, Kwonjoon L, Zhuowen T. Dual contradistinctive generative autoencoder / In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, P 823–832.
17. Kevin F, Lisa B. Soros, Olaf W. Clipdraw: Exploring text-to-drawing synthesis through language image encoders, 2021; 2106.14843.
18. Ian J., Jean P, Mehdi M, Bing X, David W, Sherjil O, Aaron C, Yoshua B. Generative adversarial networks. CoRR, 2014.
19. Martin H, Hubert R, Thomas U, Bernhard N, Sepp H. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Adv. Neural Inform. Process. Syst., 2017. P. 6626–6637.
20. Jonathan H, Chitwan S, William C, David J. Fleet, Mohammad N, Tim S. Cascaded diffusion models for high fidelity image generation. CoRR, 2021; 2106.15282.
21. Larochelle, H. The neural autoregressive distribution estimator / Larochelle, H. and Murray, I., 2011. International Conference on Artificial Intelligence and Statistics, pp. 29-37.
22. Goodfellow I, Pouget-Abadie J, Mirza M, et al. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.
23. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434.

24. Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.
25. Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).
26. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 5907-5915).
27. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).
28. 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).
29. Antipov, G., Baccouche, M., & Dugelay, J. L. (2017, September). Face aging with conditional generative adversarial networks. In 2017 IEEE international conference on image processing (ICIP) (pp. 2089-2093). IEEE.
30. Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. 2019b. Wide-context semantic image extrapolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp 1399–1408).
31. Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. 2019b. Diversity-sensitive conditional generative adversarial networks. arXiv preprint arXiv:1901.09024 (2019).
32. Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. 2020. Contextual residual aggregation for high-resolution image inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp 7508–7517).
33. Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018b. Generative image inpainting with contextual attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp 5505–5514).

34. Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2019. Free-form image inpainting with gated convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp 4471–4480)
35. Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. 2018a. Crafting a toolchain for image restoration by deep reinforcement learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp 2443–2452).
36. UI/UX Case Study з цільової сторінки AI [Електронний ресурс]. – Режим доступу: <https://shorturl.at/gizAC> (дата останнього звернення: 18.11.2023). – Назва з екрана.
37. Про революцію в дизайні завдяки MidJourney і AI [Електронний ресурс]. – Режим доступу: <https://diroх.com/post/how-midjourney-ai-image-generators-revolutionize-design> (дата останнього звернення: 18.11.2023). – Назва з екрана.
38. Про використання ШІ в графічному дизайні [Електронний ресурс]. – Режим доступу: <https://serokell.io/blog/how-is-ai-used-in-graphic-design> (дата останнього звернення: 18.11.2023). – Назва з екрана.
39. Про інтегрований у Photoshop функціонал ШІ Firefly [Електронний ресурс]. – Режим доступу: <https://www.marktechpost.com/2023/05/26/adobe-has-integrated-firefly-directly-into-photoshop-marrying-the-speed-and-ease-of-generative-ai-with-the-power-and-precision-of-photoshop/> (дата останнього звернення: 18.11.2023). – Назва з екрана.
40. Про ШІ інструменти для product дизайну [Електронний ресурс]. – Режим доступу: <https://uxplanet.org/7-ai-tools-every-product-designer-should-try-c12b00f165e1> (дата останнього звернення: 18.11.2023). – Назва з екрана.
41. Dall-E 3.0 в product дизайні [Електронний ресурс]. – Режим доступу: <https://uxplanet.org/dall-e-3-ai-image-generator-for-product-designers-5bee4c0a97d5> (дата останнього звернення: 18.11.2023). – Назва з екрана.
42. Генератори ШІ для креативної індустрії [Електронний ресурс]. – Режим доступу: <https://gcmori.medium.com/how-creatives-can-use-ai-image-generators-537f2aefdc82> (дата останнього звернення: 18.11.2023). – Назва з екрана.

43. Кращі генератори ШІ [Електронний ресурс]. – Режим доступу: <https://www.movavi.com/learning-portal/best-ai-image-generator.html> (дата останнього звернення: 18.11.2023). – Назва з екрана.

44. ШІ в мультимедіа контенті та маркетинг галузі [Електронний ресурс]. – Режим доступу: <https://aicontentfy.com/en/blog/ai-generated-content-for-video-marketing-and-multimedia-content> (дата останнього звернення: 18.11.2023). – Назва з екрана.

45. Щодо prompt-запитів, які допомагають опанувати MidJourney в галузі web-design [Електронний ресурс]. – Режим доступу: <https://www.saplingcorp.com/journals/20/midjourney-web-design-prompts> <https://gcmori.medium.com/how-creatives-can-use-ai-image-generators-537f2aefdc82> (дата останнього звернення: 18.11.2023). – Назва з екрана.

46. Використання Adobe Firefly в product дизайні [Електронний ресурс]. – Режим доступу: <https://uxplanet.org/using-adobe-firefly-to-make-in-house-product-design-unboring-ab6eeb29ca08> (дата останнього звернення: 18.11.2023). – Назва з екрана.

47. Використання Adobe Firefly product дизайні галузі загалом [Електронний ресурс]. – Режим доступу: <https://bootcamp.uxdesign.cc/how-adobes-new-firefly-will-change-design-7c03f0a36096> (дата останнього звернення: 18.11.2023). – Назва з екрана.

48. Зміни, що приносить генеративний ШІ в креативній індустрії [Електронний ресурс]. – Режим доступу: <https://hbr.org/2022/11/how-generative-ai-is-changing-creative-work> (дата останнього звернення: 18.11.2023). – Назва з екрана.

49. Теоретична інформація щодо платформ розробки веб-ресурсу [Електронний ресурс]. – Режим доступу: <https://bootcamp.uxdesign.cc/how-adobes-new-firefly-will-change-design-7c03f0a36096> (дата останнього звернення: 18.11.2023). – Назва з екрана.

50. Інформація щодо платформи Squarespace [Електронний ресурс]. – Режим доступу: <https://www.tooltester.com/en/reviews/squarespace-review/> (дата останнього звернення: 18.11.2023). – Назва з екрана.

Результати створення веб-ресурсу

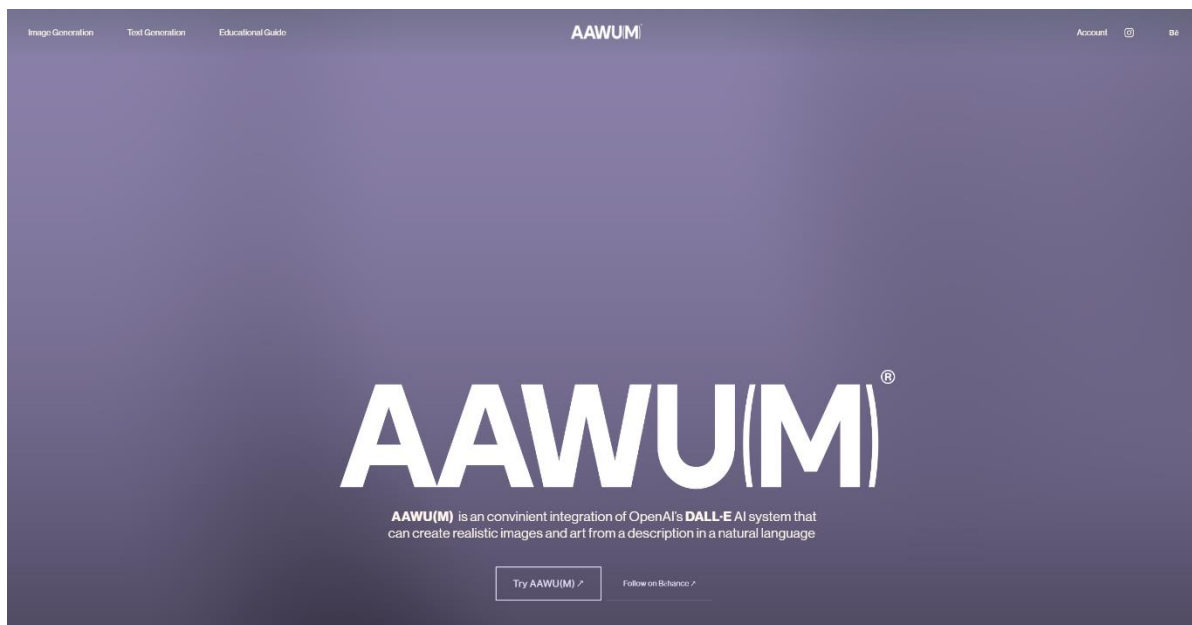


Рис. А.1. Головна сторінка

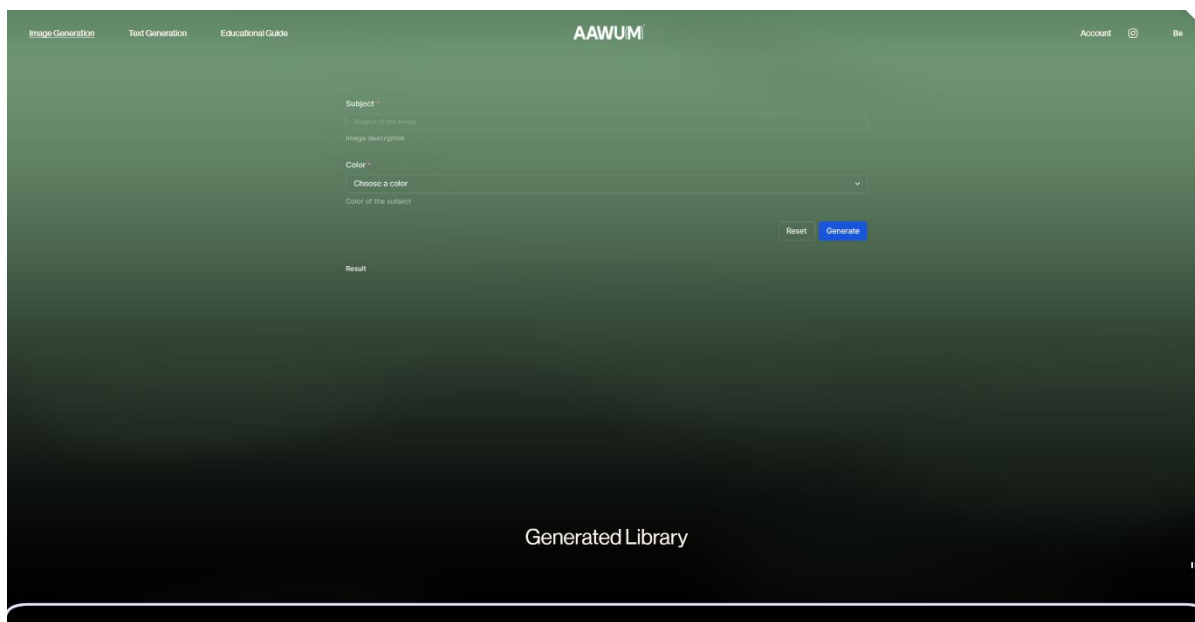


Рис. А.2. Сторінка генерації зображення

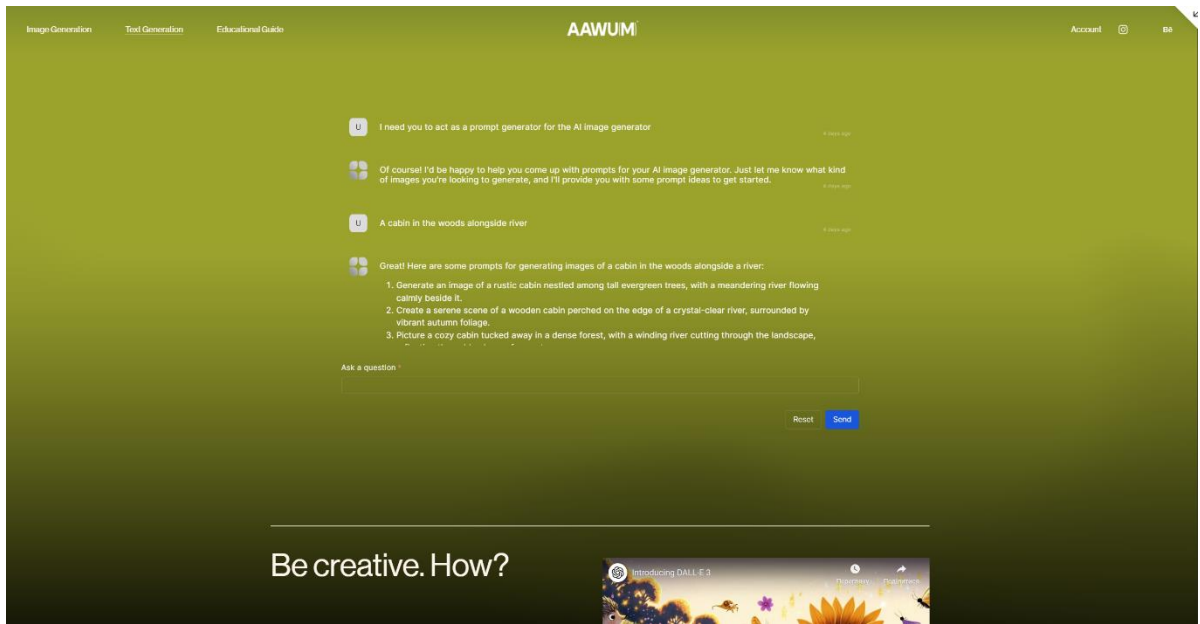


Рис. А.3. Сторінка генерації тексту

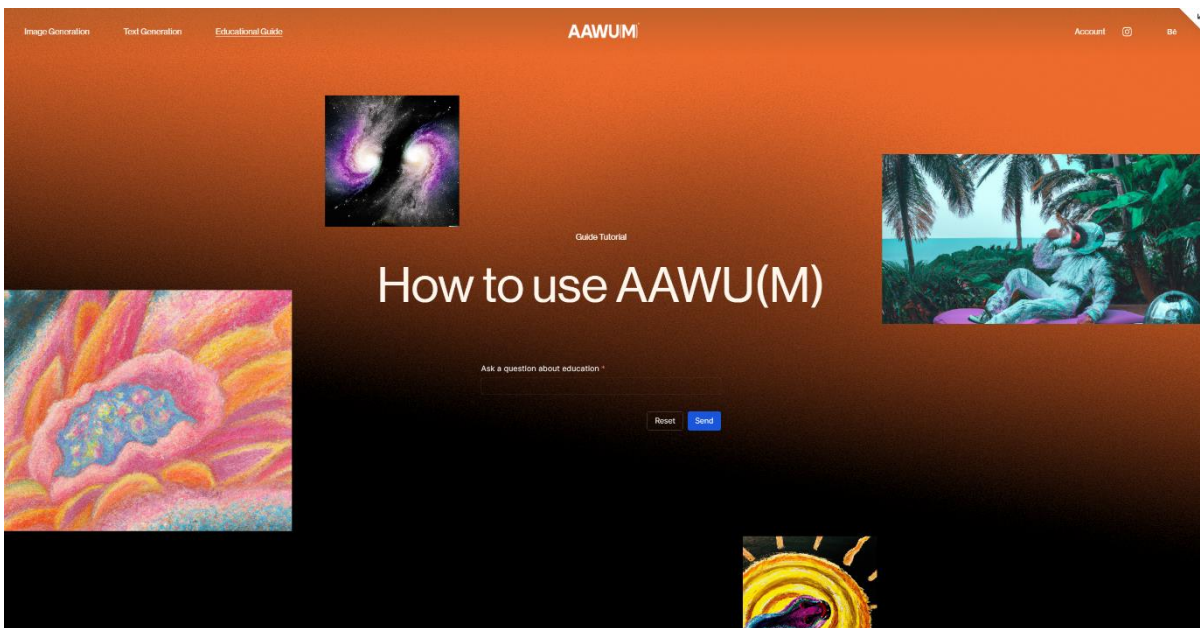


Рис. А.4. Сторінка з Education ботом