ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Створення LoRA моделі для генерації зображень за допомогою текстових запитів»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Вадим ПАНЧЕНКО
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. ІСД-42
Вадим ПАНЧЕНКО
Ім'я, ПРІЗВИЩЕ

Керівник: Юлія КАГРАМАНОВА
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем
Ступінь вищої освіти Бакалавр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІПЗАС
_____ Каміла СТОРЧАК
«____» _____20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Панченку Вадиму Юрійовичу

(прізвище, ім'я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Створення LoRA моделі для генерації зображень за допомогою текстових запитів
керівник кваліфікаційної роботи Юлія Каграманова
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024р. №36

2. Строк подання кваліфікаційної роботи «31» травня 2024р.

3. Вихідні дані до кваліфікаційної роботи:
Середовище генеразії зображень AUTOMATIC1111_Stable Diffusion web UI
Середовище навчання моделей ШІ khoya_ss UI
Науково-технічна література з питань, пов'язаних зі штучним інтелектом

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Здійснити аналіз основних методів генерації зображень
2. Провести аналіз LoRA
3. Здійснити аналіз середовищ для генерації зображень та навчання моделей ШІ
4. Провести аналіз та підготовку датасету для навчання LoRA-моделі
5. Створити LoRA-модель
6. Провести інтеграцію LoRA-моделі у середовище для генерації зображень та провести її тестування

5. Ілюстративний матеріал: *презентація*

6. Дата видачі завдання: «27» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	15.03.24 – 30.03.24	
2	Аналіз процесу генерації зображень	01.04.24 – 14.04.24	
3	Аналіз LoRA	15.04.24 – 22.04.24	
4	Встановлення та використання середовищ для генерації зображень та навчання моделей ШІ	23.04.24 – 27.04.24	
5	Розробка датасету для навчання LoRA-моделі	28.04.24 – 13.05.24	
6	Навчання, впровадження та тестування створеної моделі	14.05.24 – 19.05.24	
7	Розробка демонстраційних матеріалів (презентація)	20.05.24 – 25.05.24	

Здобувач вищої освіти _____ Вадим ПАНЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Юлія КАГРАМАНОВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавр: 64 стор., 56 рис., 20 джерел.

Мета роботи – проаналізувати та створити LoRA-модель для генерації зображень, навченої на зображеннях та текстових описах з датасету.

Об'єкт дослідження – процес створення LoRA-моделі, яка здатна трансформувати сучасні методи генерації зображень за текстовими запитами, відкриваючи нові горизонти у сфері персоналізованого візуального контенту.

Предмет дослідження – порівняння методів генерації, архітектуру LoRA-моделей, підготовку даних, навчання, впровадження та оцінку якості зображень.

Короткий зміст роботи: У роботі досліджено створення моделі LoRA для генерації зображень за текстовими запитами. Визначено, що існуючі методи генерації стикаються з обмеженнями у реалістичності та потребують значних ресурсів. LoRA-моделі забезпечують кращу якість зображень, ефективніше використовують ресурси та потребують менше даних для навчання.

Робота включає аналіз існуючих методів генерації, опис архітектури та навчання LoRA-моделі, підготовку даних, та оцінку якості згенерованих зображень. Дослідження показує перспективність LoRA-моделей у медіа, рекламі, ігровій індустрії, дизайні та медицині, відкриваючи нові можливості для створення персоналізованого візуального контенту.

КЛЮЧОВІ СЛОВА: LORA-МОДЕЛЬ, СТВОРЕННЯ МОДЕЛІ, ГЕНЕРАЦІЯ ЗОБРАЖЕНЬ, НАВЧАННЯ LORA-МОДЕЛІ, МЕТОДИ ГЕНЕРАЦІЇ, LORA, ВІЗУАЛЬНИЙ КОНТЕНТ, ТЕКСТОВІ ЗАПИТИ, НАВЧАННЯ МОДЕЛІ, ДАТАСЕТ.

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня бакалавра

Направляється здобувач Панченко В.Ю. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)
на тему: «Створення LoRA моделі для генерації зображень за допомогою текстових запитів».

Кваліфікаційна робота і рецензія додаються.

Директор ННІІТ _____ Андрій БОНДАРЧУК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Вадим Юрійович Панченко у дипломній роботі надає комплексне бачення використання LoRA-моделей для генерації зображень. Він приводить практичні рекомендації щодо вибору оптимальних моделей генерації, таких як Stable Diffusion, та детально пояснює принципи їх роботи на зрозумілих прикладах. Також підкреслює важливість детальної підготовки датасету та налаштувань моделей, що дозволяє досягти кращих результатів у генерації зображень. Його робота демонструє значний потенціал LoRA-моделей для різних сфер діяльності, включаючи мистецтво, дизайн та маркетинг.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Панченко В.Ю на оцінку «відмінно» та присвоїти йому(їй) кваліфікацію бакалавр з інформаційних систем та технологій.

Керівник кваліфікаційної роботи _____ Юлія КАГРАМАНОВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

«____»_____2024 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Панченко В.Ю. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою Інженерії програмного забезпечення автоматизованих систем _____ Каміла СТОРЧАК
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня бакалавра

здобувача(ки) вищої освіти Панченко Вадим Юрійович
(прізвище, ім'я, по батькові)
на тему: «Створення LoRA моделі для генерації зображень за допомогою текстових запитів»

Актуальність.
Актуальність цієї теми особливо важлива в сучасних умовах, де технологічний прогрес і зміни в поведінці споживачів створюють нові вимоги до візуального контенту. Соціальні мережі, електронна комерція та мультимедійні платформи все більше потребують унікальних зображень. В такому контексті LoRA-модель для генерації зображень на основі текстових запитів має значний потенціал, що відповідає сучасним викликам та потребам.

Позитивні сторони.
1. Дипломна робота демонструє детальний аналіз методів генерації зображень (GAN, VAE, Stable Diffusion), що сприяє кращому розумінню цих технологій.
2. Створена LoRA-модель успішно генерує зображення обличчя, показуючи непогані результати навіть з обмеженим датасетом з 16 зображень.
3. Робота підкреслює потенціал LoRA-моделей для мистецтва, дизайну та маркетингу, відображаючи сучасні технологічні тренди і потреби ринку.

Недоліки.
1. Навчання LoRA-моделі проводилось на невеликому датасеті з 16 зображень, що може впливати на якість результатів.
2. Математичні аспекти LoRA-моделей залишаються складними для розуміння, що може утруднити засвоєння матеріалу для новачків.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи бакалаврської.

Висновок: *кваліфікаційна робота на здобуття ступеня бакалавра заслуговує оцінку «відмінно», а здобувач Панченко В.Ю. заслуговує присвоєння кваліфікації: бакалавр з інформаційних систем та технологій*

Рецензент: _____ _____
науковий ступінь, вчене звання *підпис* *Ім'я, ПРІЗВИЩЕ*

ЗМІСТ

Стор.

ВСТУП

Постановка проблеми. Зростаючий попит на генерацію зображень за текстовими запитами вимагає унікального та персоналізованого контенту, але існуючі методи стикаються з обмеженнями, такими як недостатня реалістичність та висока потреба в обчислювальних ресурсах і даних. LoRA-моделі (Low-Rank Adaptation) можуть вирішити ці проблеми, забезпечуючи більшу реалістичність і різноманітність зображень завдяки кращому розумінню текстових описів. Вони також оптимізовані для ефективного використання ресурсів і потребують менше даних для навчання, що робить їх більш придатними для широкого застосування. Отже, використання LoRA-моделей для генерації зображень за текстовими запитами є перспективним напрямом дослідження, здатним покращити якість та ефективність генерованого контенту.

Актуальність теми. Зростаючий попит на генерацію зображень зумовлений розвитком технологій і змінами у споживчих звичках. Соціальні мережі, електронна комерція та мультимедійні платформи потребують унікального візуального контенту. У цьому контексті LoRA-модель для генерації зображень за текстовими запитами є дуже перспективною.

Ця технологія може застосовуватися в різних сферах. У медіа та рекламі вона автоматизує створення рекламних матеріалів на основі текстових описів. У ігровій індустрії LoRA-модель допомагає створювати нових персонажів і локації. У дизайні та моді вона швидко генерує концептуальні зображення нових виробів. У медичній сфері ця модель створює візуальні репрезентації медичних даних, допомагаючи зрозуміло представляти інформацію.

Отже, розробка LoRA-моделей для генерації зображень за текстовими запитами відкриває нові можливості для створення персоналізованого і якісного візуального контенту, що відповідає сучасним потребам.

Мета та завдання дослідження. Метою цього дослідження є створення моделі LoRA, яка базується на великій моделі генерації зображень та наборі даних,

що складається з фотографій та текстових файлів з їхніми описами, з метою навчання моделі.

Задля реалізації мети дослідження, було сформульовано наступні завдання:

а) Порівняти та проаналізувати існуючі методи генерації зображень, зокрема GAN, VAE та моделі дифузії, з метою виявлення їхніх переваг та недоліків.
б) Опис архітектури та принципи роботи LoRA-моделей для генерації зображень.
в) Визначення ключових переваг та оцінка обмеженнь LoRA-моделей.
г) Опис середовища для генерації зображень та середовища навчання LoRA-моделей.
д) Обґрунтувати вибір конкретної архітектури LoRA для дослідження та провести її детальний аналіз.
е) Підготувати дані для навчання моделі, включаючи збір та підготовку текстових та візуальних даних.
ж) Провести процес навчання LoRA-моделі для генерації зображень.
з) Провести якісну оцінку згенерованих зображень.
и) Узагальнити результати дослідження та виокремити ключові висновки щодо ефективності LoRA-моделі.
к) Пояснити практичну користь та можливості використання розробленої моделі, а також визначити перспективні напрямки для подальшого розвитку дослідження.

Об’єкт та предмет дослідження. Об'єктом дослідження є процес генерації зображень за допомогою текстових запитів, та створення LoRA-моделі.

Предметом дослідження є розробка та аналіз LoRA-моделей для створення зображень з текстових описів. Дослідження включає порівняння методів генерації, опис архітектури LoRA-моделей, підготовку даних, процес навчання, впровадження моделі та оцінку якості зображень.

Дослідження також включатиме тестування створеної моделей, аналіз результатів та пошук можливостей для вдосконалення. Метою є створення LoRA-моделі, здатної генерувати високоякісні зображення за текстовими описами.

Методи дослідження. Для дослідження теми “Створення LoRA-моделі для генерації зображень за допомогою текстових запитів” буде проведено:

а) Порівняльний аналіз: Порівняння методів генерації зображень (GAN, VAE, моделі дифузії) для виявлення їх переваг і недоліків.
б) Огляд літератури та аналіз документів: Аналіз наукових статей та технічної документації для розуміння архітектури та принципів роботи LoRA-моделей.
в) Емпіричний аналіз: Визначення переваг та обмежень LoRA-моделей.
г) Моделювання та симуляція: Опис середовища для генерації зображень та навчання LoRA-моделей.
д) Аналіз архітектури : Вибір і детальний аналіз конкретної архітектури LoRA для навчання моделі.
е) Методи підготовки даних: Збір, підготовка та обробка текстових і візуальних даних для навчання моделі.
ж) Навчання моделі: Проведення навчання LoRA-моделі на підготовлених даних.
з) Використання моделі: Повести впровадження створеної моделі у середовище для генерації зображень.
и) Методи якісної оцінки: Оцінка якості згенерованих зображень через аналіз та порівняльні тести.
к) Узагальнення та аналіз результатів: Систематизація результатів та формування висновків щодо ефективності LoRA-моделі.
л) Оцінка практичної користі: Аналіз можливостей практичного застосування моделі та визначення напрямків для подальших досліджень.

Апробація результатів дослідження. Попередні результати роботи були апробовані на V Міжнародній науково-технічній конференції «Сучасний стан та перспективи IoT», яка проходила 18 квітня 2024 року. Тези на тему «Вплив відеокарт на ШІ» та «Оцінка якості моделі ШІ» було опубліковано у збірнику, присвяченому цій конференції.

1 МЕТОДИ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ ТА LoRA-МОДЕЛІ: АНАЛІЗ І ПЕРЕВАГИ

1.1 Огляд існуючих методів генерації зображень та їх порівняння

Цей підрозділ має на меті поверхнево дослідити та порівняти основні методи генерації зображень. Зосередимося на розумінні процесів генерації зображень, їхніх відмінностей, потенційних переваг та недоліків.

Для початку треба зазначити, що на даний момент є досить велика кількість методів генерації зображень, а також кожного місяця з’являються нові. Тому я зосереджуся на основних та найбільш популярних, таких як:

1. Генеративні змагальні мережі (GAN)
2. Автокодери з варіаціями (VAE)
3. Дифузійні моделі

1. Почнемо з GAN. Один з типів архітектури глибокого навчання називається генеративною змагальною мережею (GAN). Ця модель має декілька варіантів використання в різних галузях, але ми зосередимося саме на зображенях. На основі заданого навчального набору даних GAN навчає дві нейронні мережі, які мають конкурувати одна з одною, щоб створювати більш оригінальні нові дані. Ви можете створити нові зображення з існуючої бази даних зображень. Оскільки GAN тренує дві різні мережі і протиставляє їх одна одній, цей процес генерації зображень відомий як змагальна мережа. Застосовуючи якомога більше модифікацій до зразка вхідних даних, одна мережа (генеративна) створює нові дані, а інша мережа (прогнозуюча) намагається спрогнозувати, чи відповідають отримані дані вихідному набору даних. Інакше кажучи, прогнозуюча мережа встановлює достовірність згенерованих даних. Поки прогнозуюча мережа не зможе відрізнити фальшиві дані від справжніх, система продовжуватиме створювати кращі та новіші версії фальшивих даних.

Використовуючи текстові підказки або змінюючи вже існуючі зображення, генеративні змагальні мережі створюють реалістичні зображення. У відеоіграх та інших цифрових розвагах вони можуть допомогти у створенні реалістичних і захоплюючих візуальних вражень.

Крім того, GAN здатний редагувати зображення, наприклад, перетворювати чорно-біле зображення на кольорове або зображення з низькою роздільною здатністю на зображення з високою роздільною здатністю. Для анімації та відео він також може створювати реалістичні обличчя, персонажів і тварин.

Але як проходить сам процес навчання? Весь обчислювальний процес базується на складному математичному рівнянні, це спрощений опис:

— після обробки набору даних для навчання генеративна мережа визначає їх атрибути;
— крім того, нейронна прогнозуюча мережа також досліджує навчальний набір даних і самостійно визначає атрибути;
— додаючи шум (або випадкові зміни) до певних атрибутів, генеративна мережа модифікує деякі атрибути даних;
— прогнозуюча мережа отримує оновлені дані від генеративної;
— прогнозуюча мережа визначає ймовірність того, що отримані дані, які вона отримала від генеративної мережі, є частиною початкового набору даних;
— щоб зменшити рандомізацію вектора шуму в наступному циклі, прогнозуюча мережа надає генеративній мережі певні вказівки.

Прогнозуюча мережа прагне мінімізувати ймовірність помилки, тоді як генеративна мережа прагне зробити все навпаки та максимізувати ймовірність помилки. Генеративна і прогнозуюча мережі постійно змінюються і стикаються один з одним під час навчальних ітерацій, поки не досягнуть стану рівноваги. У цьому стані прогнозуюча мережа більше не може ідентифікувати синтезовані дані. Процес навчання завершено. Для кращого розуміння можете звернути увагу на зображення схеми моделі GAN (рис 1.1).

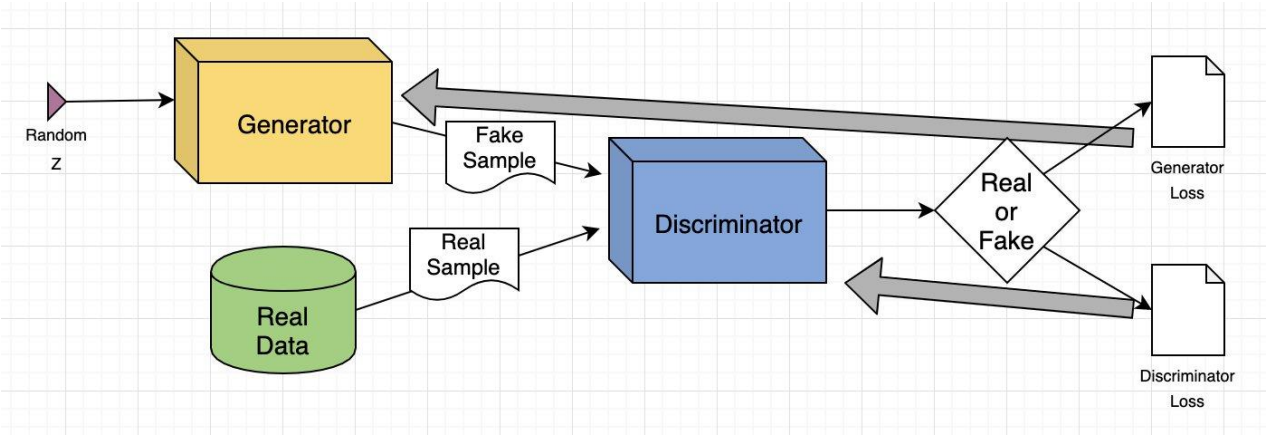


Рис. 1.1 Схема роботи методу генерації зображень GAN

На мою думку, це був досить зрозумілий приклад, але давайте розглянемо наведену вище інформацію на прикладі того, як модель GAN модифікує зображення.

Уявіть, що GAN намагається змінити людське обличчя на вхідному зображенні. Форми вух або очей, наприклад, можна вважати атрибутами. Припустимо, що генеративна мережа модифікує вихідні зображення, додаючи до них сонцезахисні окуляри. Прогнозуюча мережа отримує набір зображень, деякі з яких є згенерованими зображеннями, які були відредаговані для додавання сонцезахисних окулярів, а деякі - реальними людьми в сонцезахисних окулярах.

Генеративна мережа змінює свої параметри, щоб створити ще більш якісні фальшиві зображення, якщо прогнозуюча мережа здатна відрізнити фальшиві від справжніх. Прогнозуюча мережа змінює свої параметри, якщо генеративна мережа генерує зображення, які її обманюють. Обидві мережі вдосконалюються через конкуренцію, поки не досягнуть рівноваги.

Цей метод має як значущі плюси так і деяки мінуси, але їх я визначу у процесі порівння з іншими методами генерації зображень до яких ми зараз перейдемо.

2. Другим методом генерації зображень який я виділив є VAE. VAE - це ще один тип генеративної моделі ШІ, що використовується для синтезу зображень. Серед таких моделей вони займають особливе місце, тому що поєднують в собі елементи автоенкодера та байєсівських методів. VAE є одним із перших методів глибокого навчання, які дозволяють генерувати зображення з латентного простору.

Але перш ніж почати обговорювати VAE, треба розібратися, що з себе представляють латентний простір та кодувальник і декодувальник, про які пійде мова при поясненні принципу роботи VAE:

— кодувальник - це компонент моделі, який перетворює вхідні дані в інше представлення;

— декодувальник - це частина моделі, яка відновлює вихідні дані з прихованого представлення (отриманого, наприклад, від кодувальника).

— латентний простір - це вбудовування набору елементів у множину, де елементи, які схожі один на одного, знаходяться ближче один до одного. Його також називають простором прихованих об'єктів або простором вбудовування.

Для кращого розуміння латентного простору зіставимо цей принцип із тим, як люди сприймають світ навколо.

Наприклад, коли ми гуляємо у лісі і бачимо дерево, ми не фіксуємо увагу на кожній маленькій деталі цієї рослини, а зосереджуємося на загальній формі, типі та розмірі. Ми миттєво впізнаємо дерево, не занурюючись у надмірну інформацію (Рис. 1.2).

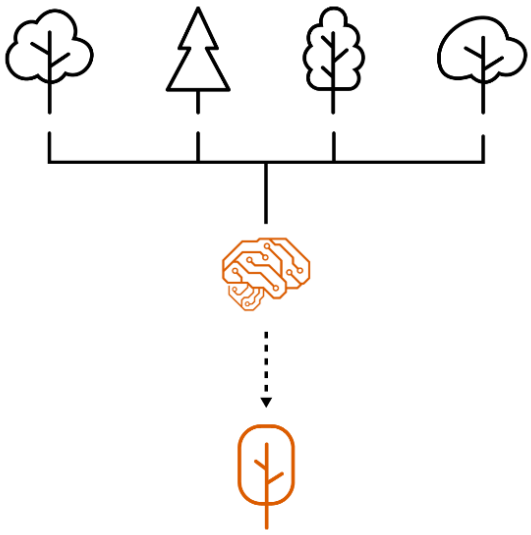


Рис. 1.2 Об'єднання декількох дерев в одне відображення

Подібно до цього, латентний простір має на меті дати комп'ютеру стисле уявлення про дерево. Не називаючи конкретно кожну гілку, він привертає увагу до основних характеристик дерева, таких як наявність гілок, структура стовбура та форма крони.

Іншими словами, це просто стисле представлення даних (рис 1.3), в якому пов'язані точки даних згруповані разом у просторі.

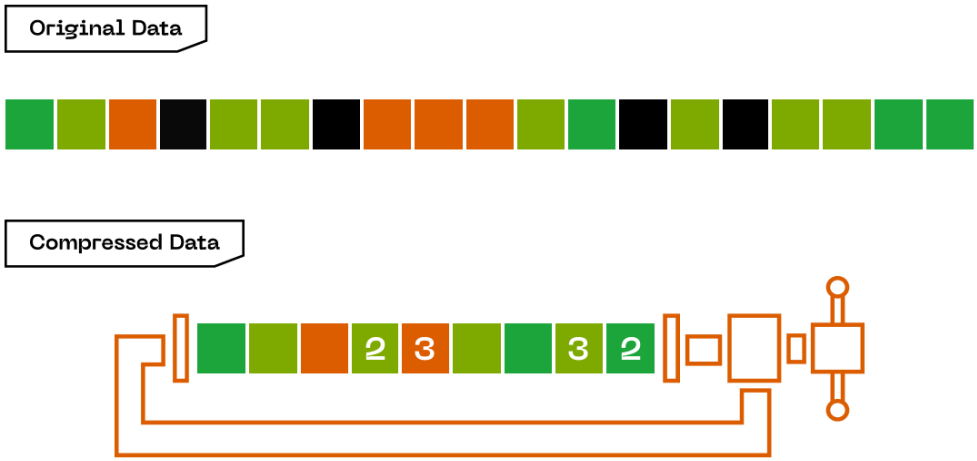


Рис. 1.3 Як перетворюються дані

Використовуючи множини, кластеризацію та інші методи, ми можемо аналізувати дані в латентному просторі і, таким чином, розуміти закономірності або структурну схожість між точками даних.

Латентний простір корисний для вивчення функцій даних і знаходження більш зручного для аналізу представлення даних.

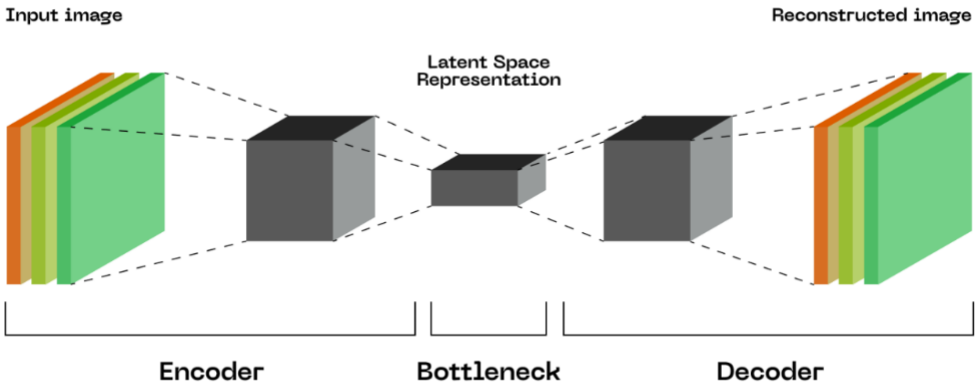


Рис. 1.4 Взаємодія кодувальника і декодувальника

Як бачимо, зображення торкається таких понять, як кодувальник і декодувальник (рис. 1.4). Однак, як вони функціонують?

Ціль кодувальника - створити вектори в латентному просторі, де схожі об'єкти будуть розташовані ближче один до одного, а різні - далі. Це сприяє формуванню чітких кластерів, де схожі об'єкти групуються разом.

Уявімо, що у нас є кодувальник, який вміє обробляти зображення дерев. Якщо ми надамо зображення ялини, він розташує її вектор поруч з векторами, що відповідають хвойним деревам; якщо ми надамо зображення дуба, він розташує вектор зображення в іншій області латентного простору, поруч з векторами, що відповідають листяним деревам (рис. 1.5).

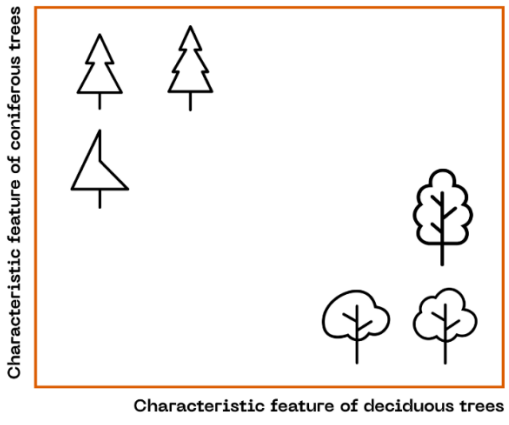


Рис. 1.5 Схема розбиття кодувальником дерев на різні групи за їх параметрами

Після навчання декодер може відновлювати оригінальні об'єкти з латентного вектора з високою розмірністю. Важливо пам'ятати, що декодер можна використовувати не лише для відновлення оригінальних об'єктів, але й для генерування абсолютно нових даних. Для цього достатньо надати йому латентне представлення об'єктів, які не були включені в навчальний набір даних.

Коли ми розібралися, що таке латентний простір, кодувальник і декодувальник, та як вони працюють, - можна перейти до методу генерації зображень VAE.

VAE складається з двох основних модулів: кодувальника і декодувальника. Кодувальник приймає вхідне зображення і перетворює його в латентний простір, в якому зображення може бути представлене у вигляді вектора. Декодувальник, в свою чергу, приймає цей вектор з латентного простору і відтворює вихідне зображення. Однак, основна відмінність VAE від класичних автоенкодерів полягає в тому, що його кодувальник вивчає розподіл згенерованих латентних векторів, а не конкретні значення. Іншими словами, VAE намагається навчитися параметризувати розподіл у латентному просторі, що дозволяє генерувати нові зображення шляхом семплювання з цього розподілу.

Коли VAE навчається, він робить це через взаємодію двох основних частин: кодувальної і декодувальної мереж (рис. 1.6). Ось як працює цей процес:

— кодувальник: ця мережа отримує вхідні дані і перетворює їх у вектор, який представляє атрибути, такі як середнє значення і дисперсія. Ці атрибути можна розглядати як код вхідних даних;

— декодувальник: ця мережа отримує вектор з кодувальної мережі і використовує його для генерації нового зображення. Вона намагається відновити вхідні дані з цього вектора;

— додавання шуму: тепер важливий момент. Ми додаємо до коду з кодувальника невеликий шум, що робить його трохи випадковим. Це робить кожен вектор унікальним і може сприяти більшій різноманітності в генерованих зображеннях;

— повернення до декодувальника: змінений код із шумом подається на вхід декодувальника, який намагається відновити оригінальні дані;
— порівняння з оригіналом: декодувальник порівнює згенероване зображення з оригінальним. Його завдання - зрозуміти, наскільки воно схоже на вихідні дані;
— оцінка втрат: на основі порівняння декодувальник оцінює, наскільки добре він зробив свою роботу. Ця оцінка допомагає мережі коригувати параметри так, щоб вона генерувала найкращі зображення;
— повторення: цей процес повторюється декілька разів для всього набору даних, поки декодувальник не навчиться генерувати реалістичні зображення, а кодувальник - ефективно їх кодувати.

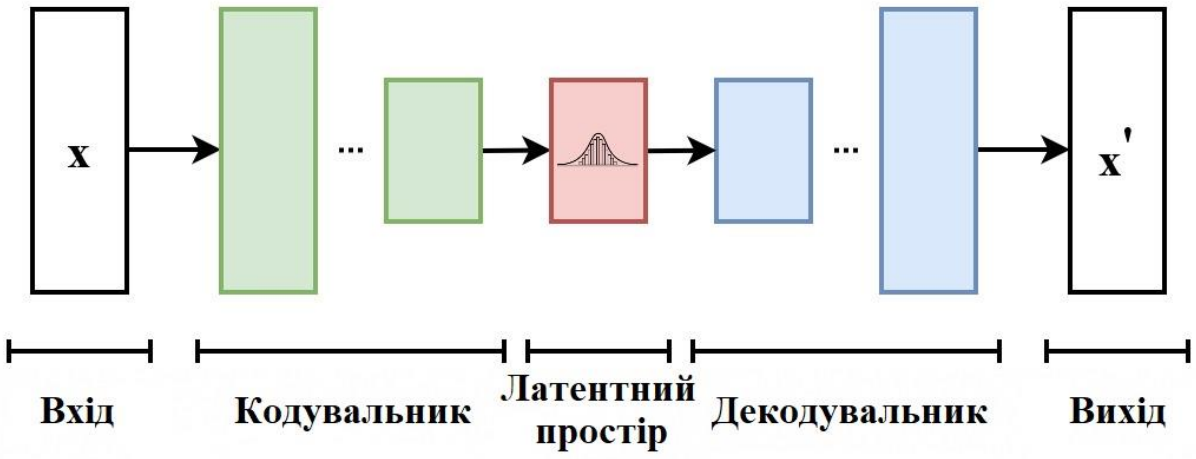


Рис. 1.6 Схема роботи методу генерації зображень GAN

Загальний процес навчання VAE може бути складним через необхідність оптимізації коду і навчання згаданого розподілу. Однак, за допомогою відповідної архітектури та оптимізаційних методів, VAE може виявлятися ефективним і потужним інструментом для генерації зображень.

3. Останнім та найбільш важливим для цієї дипломної роботи методом генерації зображень, який я виділив, є дифузійні моделі (Stable Diffusion). На них ми зосередимося трошки більше.

Модель генеративного штучного інтелекту під назвою Stable Diffusion створює оригінальні, фотореалістичні зображення у відповідь на текстові та графічні підказки. Її перший запуск відбувся у 2022 році. Модель можна використовувати для створення анімації та відео на додаток до зображень. Модель використовує латентний простір, про який ми згадували перед визначення VAE, і базується на технології дифузії. Про процес генерації зображень я розповім згодом.

Користувачі можуть легко вивчити і використовувати цю модель завдяки великій кількості ресурсів і навчальних посібників, що надаються активною спільнотою Stable Diffusion.

Модель Stable Diffusion використовується як на різноманітних веб-сайтах для генерації зображень, так і у вигляді десктопного веб-інтерфейсу. Веб-сайти, що забезпечують генерацію зображень за методом Stable Diffusion, мають простий у використанні інтерфейс, що полегшує початок роботи для користувачів низької кваліфікації. Деякі з цих сайтів спрощують створення зображень з необхідними характеристиками, надаючи заздалегідь створені шаблони і налаштування, але зображення згенерові такими сервісами досить посередні. Деякі веб-сайти, такі як славнозвістний Midjourney, мають ліміт на кількість зображень, які можна згенерувати, або потребуть коштовної підписки, при цьому надають користувачам більше свободи та контролю над процесом створення зображень, дозволяючи змінювати різні параметри моделі. Користувачі мають ще більше можливостей для створення зображень при використанні безкоштовного Open Source веб-інтерфейсу Stable Diffusion для десктопів. За допомогою цього інтерфейсу можна отримати доступ до всіх функцій моделі та змінити її відповідно до своїх вимог. Оскільки метод генерації зображень Stable Diffusion є легкодоступним і простим у використанні, це має велике значення. Відеокарти, що втробляють для споживачів, можуть його запускати. Професійні дизайнери, художники та дослідники, які потребують найбільшої гнучкості та контролю над процесом створення зображень,

знайдуть Stable Diffusion ідеальним інструментом. Він дозволяє створювати фотореалістичні зображення, що майже ідентичні реальним. Більш детальний опис цього веб-інтерфейсу буде згодом у наступних розділах.

Процес генерації зображень за допомогою Stable Diffusion можна розділити на наступні етапи:

— підготовка даних: першим кроком є створення навчального набору даних з високоякісними зображеннями. На цьому наборі даних тренується модель дифузії;

— навчання дифузійної моделі: дифузійна модель навчається на наборі даних, поступово додаючи шум до зображень та навчаючись зворотньому процесу видалення шуму;

— введення текстового запиту: користувач надає текстовий опис зображення як підказку для генерації зображення після навчання дифузійної моделі;

— початкова генерація зашумленого зображення: на основі текстової підказки створюється початкове зашумлене зображення, що є основою для процесу дифузії;

— ітеративний процес дифузії: використовуючи текстову підказку як орієнтир, модель дифузії вдається до вихідного зашумленого зображення, поступово усуваючи шум для отримання бажаного зображення. З кожною ітерацією цього процесу модель покращує і уточнює згенероване зображення;

— фінальне зображення: остаточне зображення, що має відповідати наданій текстовій підказці, є результатом ітеративного процесу дифузії.

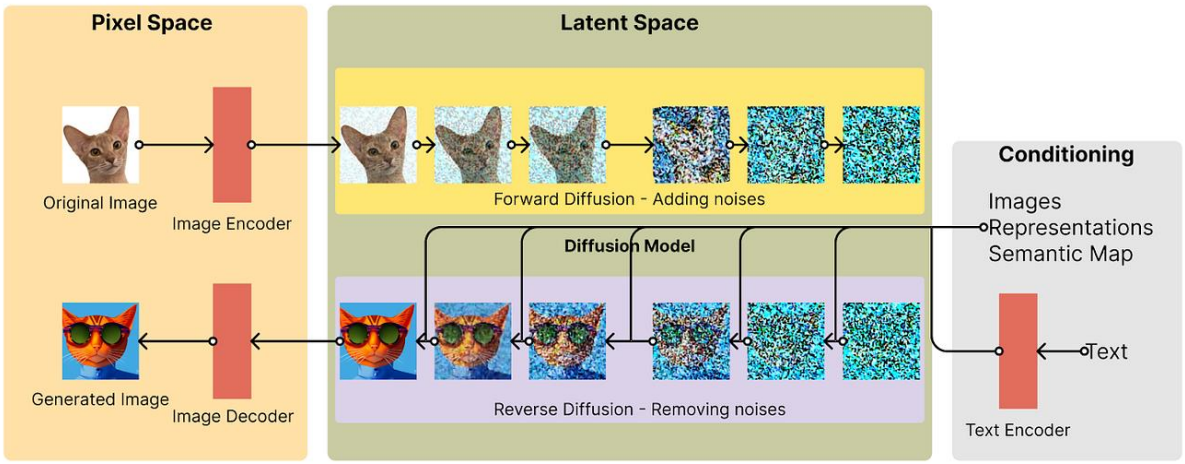


Рис. 1.7 Схема роботи методу генерації зображень Stable Diffusion

Тобто, моделі Stable Diffusion не схожі на інші моделі генерації зображень. Вони кодують зображення за допомогою гаусівського шуму, після чого відтворюють зображення, використовуючи зворотний процес дифузії та предиктор шуму (рис. 1.7).

Stable Diffusion відрізняється від інших дифузійних моделей не лише своїми технічними аспектами, але й тим, що вона не використовує піксельний простір зображення. Замість цього вона використовує латентний простір зі зниженою чіткістю.

Це пов'язано з тим, що для кольорового зображення з роздільною здатністю 512x512 існує 786 432 можливих значень. На противагу цьому, Stable Diffusion використовує стиснене зображення з 16 384 значеннями, що в 48 разів менше, і, як результат, значно зменшуються вимоги до обробки. Це пояснює, чому Stable Diffusion можна використовувати на настільному комп'ютері з відеокартою NVIDIA з 8 ГБ відеопам'яті. Менший латентний простір працює, тому що природні зображення не є випадковими. Для промальовування складних деталей, таких як очі, Stable Diffusion використовує файли варіаційного автокодера (VAE) у декодері, які ми згадували раніше.

Також хотілося б обговорити деякі можливості Stable Diffusion моделей, які ще сильніше відокремлюються на фоні інших:

— створення відео: це можна назвати створенням відео, хоча це більше схоже на створення анімації. За допомогою моделі, веб-інтерфейсу та відповідного додатку з github можна створювати анімації фото;

— генерація зображень з зображеннь: процес зміни вихідного зображення відповідно до властивостей цільового зображення або області цільового зображення - називається перетворенням зображення з зображення. У деяких випадках цей метод можна використовувати для підвищення роздільної здатності зображень;

— редагування та ретуш зображень: спеціальний інструмент під назвою inpaint замінює або редагує певні ділянки зображення. Це робить його корисним інструментом для реставрації зображень, для усунення дефектів і артефактів, або навіть для заміни частини зображення на щось абсолютно нове.

Коли ми розібралися як працють ці три методи генерації зображень, варто зазначити, що кожен тип цих моделей може бути використаний разом з LoRA. Необхідно вирішити, який саме метод генерації зображень краще підходить для роботи з LoRA-моделями:

GAN – це потужна модель, що довела свою ефективність у генерації реалістичних зображень. Однак вона потребує великої обчислювальної потужності через необхідність навчання одночасно двох моделей, складністю її розвертання, та нестабільністю такого навчання.

З іншого боку, VAN відома своєю гнучкістю та здатністю моделювати складні розподіли. Однак, часто вона має проблеми з генеруванням високоякісних вибірок.

Stable Diffusion, як було зазначено, – нещодавня інновація, що зробила революцію в галузі генеративного моделювання. Вона демонструє неперевершену продуктивність у створенні високоякісних, реалістичних та різноманітних зображень. На відміну від своїх аналогів, Stable Diffusion перевершує їх у створенні реалістичних результатів, уникаючи проблем, пов’язаних з нестабільністю навчання та складністю розвертання.

Однією з найважливіших переваг Stable Diffusion є його здатність генерувати різноманітні та реалістичні зразки. Ця модель може створювати нескінченну кількість зразків, кожен з яких має свої унікальні характеристики. Це різко контрастує з GAN, які часто страждають від обмеженої різноманітності зразків. VAN, хоча і здатні генерувати різноманітні зразки, часто мають проблеми з якістю та узгодженістю.

Інтерпретованість даних Stable Diffusion - ще один важливий аспект, в якому вона перевершує інші методи. На відміну від GAN, які, як відомо, важко інтерпретувати, стабільна дифузія забезпечує чітке розуміння процесу генерації. Ця прозорість дозволяє дослідникам більш ефективно налаштовувати та вдосконалювати модель.

На закінчення цього підрозділу варто зазначити, що хоча всі три моделі мають свої сильні сторони, але Stable Diffusion являє собою найперспективнішу модель для генерації зображень за допомогою текстових запитів. Тобто, у подальших розділах, при створенні самої LoRA-моделі, та при генерації зображень будуть використовуватися саме Stable Diffusion моделі. Вони поєднуть в собі найкращі сторони GAN і VAN, забезпечуючи при цьому унікальні переваги, які роблять їх ідеальним вибором для широкого спектра застосувань. Це робить Stable Diffusion моделлю майбутнього в галузі генерації зображень.

1.2 LoRA-моделі: огляд та принципи роботи

Після знайомства з основними методами генерації зображень, час дослідити, що ж таке LoRA-моделі та як вони пов'язані з Stable Diffusion.

Сьогодні штучний інтелект стрімко розвивається, і однією з його ключових віх є поява великих моделей. До цієї категорії належать, наприклад, згадана нами Stable Diffusion для генерації зображень, або ж LLM (Large Language Models) на кшталт GPT.

Ці потужні моделі, навчаючись на гігантських масивах даних, стають універсальними інструментами, здатними виконувати широкий спектр завдань.

Їхня гнучкість робить їх придатними для вирішення проблем різного характеру. Проте, іноді виникає нюанс: недостатня натренованість для досягнення найкращих результатів у конкретній сфері.

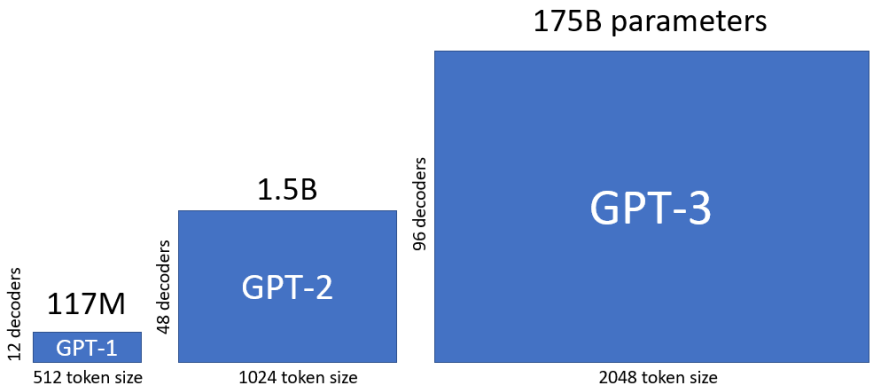


Рис. 1.9 Кількість параметрів великих мовних моделей

Незважаючи на свої 175 мільярдів параметрів (рис. 1.9), GPT-3 іноді не може задовольнити запити користувачів через те, що ґрунтується на даних, зібраних до 2022 року. Це означає, що модель не має інформації про події, явища та тренди, які з'явилися пізніше (рис. 1.10).

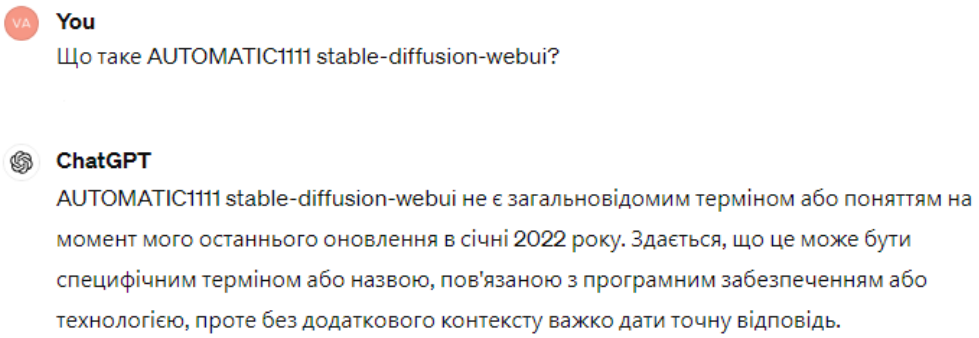


Рис. 1.10 Відповідь GPT-3 на запит, що стосується події, яка сталася після 2022 року.

Виникає питання: Як вдосконалити модель в подібних ситуаціях? Повторне навчання з нуля на новому наборі даних потребує значних ресурсів пам'яті та обчислювальних потужностей, що робить цей метод не завжди виправданим.

Замість цього можна використовувати метод LoRA (Low-Rank Adapters). Він дозволяє “точно налаштовувати” мовні моделі, такі як GPT-3, та моделі Stable Diffusion, для виконання конкретних завдань або відповідей на певні питання, точно налаштовуючи їхню поведінку під конкретні потреби.

Тонке налаштування - це метод вдосконалення моделі, що полягає в оновленні її параметрів за допомогою нових даних, які стосуються конкретного завдання. Цей процес дозволяє моделі навчатися на нюансах та особливостях цільової області, зберігаючи при цьому знання, отримані під час попереднього навчання на великому наборі даних.

Перш ніж дати чітке визначення LoRA, я хотів би пояснити принципи його роботи. Розумію, це може здатися складним завданням, адже у минулому підрозділі я намагався уникати складних математичних формул і пояснювати роботу моделей генерації зображень на зрозумілих прикладах. Однак, щоб йти далі, нам не обійтися без фундаментального розуміння лінійної алгебри та принципа роботи нейронних мереж.

Функція активації - це математична операція, що застосовується до вихідного значення нейрона. Вона вводить нелінійність, обмежує значення або розділяє класи, роблячи нейронні мережі потужнішими.

Вага - це компоненти параметрів, які безпосередньо впливають на вихідні сигнали нейронів.

Алгоритм градієнтного спуску - це ітеративний метод оптимізації, призначений для пошуку мінімуму функції.

Уявіть собі:

— Ви плануєте навчити нейронну мережу, щоб вона могла розпізнавати рукописні цифри.
— Мережа має мінімізувати помилку під час класифікації зображень.
— Мінімум помилки - це і є шуканий мінімум функції.

Лінійний шар - це набір нейронів, які застосовують до своїх даних лінійне перетворення: множення на матрицю і додавання з константою.

Функції які може виконувати лінійне перетворення:

— перетворення даних (наприклад, із зображень у вектори);

— стиснення (перекодування) даних (зменшення розмірності);

— об'єднання даних із різних джерел;

— класифікація (визначення категорії даних).

Після того, яка ми дали визначення деяким поняттям - розглянемо нейронну мережу, що складається з одного лінійного шару без функції активації.

Ця мережа приймає на вхід вектор даних $\boldsymbol{x}$ і видає на виході вектор $\boldsymbol{y}$. Розрахунок вихідного значення здійснюється за формулою:

$$y = Wx$$

Де:

W – матриця ваг;
x – вхідний вектор даних;
y – вихідний вектор.

Формула $y = Wx$ описує лінійне перетворення вхідного вектора x за допомогою матриці ваг W. Це означає, що кожне вихідне значення у є лінійною комбінацією вхідних значень x, зважених відповідно до матриці W.

Припустимо, що необхідно скоригувати роботу цієї мережі за допомогою донавчання. Для цього ваги матриці W оновлюються на ΔW (зазвичай за допомогою алгоритму градієнтного спуску).

Відомо, що після донавчання вихідне значення буде розраховуватися за наступною формулою:

$$y' = W'x = (W + \Delta W)x = y + \Delta Wx$$

Де:

y' – вихідне значення моделі після донавчання;
W' – оновлена матриця ваг після донавчання;
W – початкова матриця ваг до донавчання;
ΔW – зміна (адаптація) матриці ваг в результаті донавчання;
x – вхідне значення (вектор вхідних даних);
y – початкове вихідне значення моделі, обчислене до донавчання, тобто $y = Wx$.

Пояснення:

а) Вхідний вектор даних x: Це набір вхідних значень, які подаються на вхід мережі. Наприклад, у задачі класифікації зображень, вектор $\boldsymbol{x}$ може складатися з піксельних значень зображення.

б) Матриця ваг W: Визначає, як вхідні значення будуть перетворені у вихідні. Кожен елемент матриці ваг відповідає за "вагу" або важливість відповідного вхідного значення.

в) Обчислення $y = Wx$: Відбувається матричне множення матриці ваг W на вектор вхідних даних x. Результатом є вектор вихідних значень y.

г) Донавчання: Для покращення роботи моделі здійснюється донавчання, під час якого ваги матриці W оновлюються. Зміна ваг описується як ΔW.

д) Оновлене значення: Після донавчання, нове вихідне значення y' обчислюється за формулою $y' = (W + \Delta W)x$, що можна представити як $y + \Delta Wx$, це означає, що нове вихідне значення є сумою початкового вихідного значення $\boldsymbol{y}$ і змін, викликаних донавчанням ΔWx.

Цю зміну можна інтерпретувати як додавання до нашого шару ще одного повнозв'язного шару. Ми отримуємо спосіб модифікації роботи лінійного шару, який можна уявити як додавання до нього ще одного, окремого, повнозв'язного шару. Це дозволяє моделі краще адаптуватися до нових даних і покращувати свої передбачення.

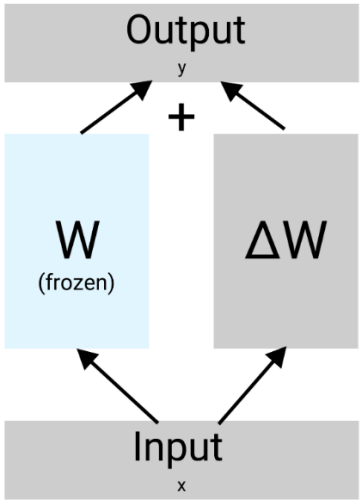


Рис. 1.11 Схема процесу навчання моделі ΔW

Замість того, щоб змінювати ваги матриці *W*, ми можемо зафіксувати (рис. 1.11) їх і навчати окрему модель *ΔW*, яка буде прогнозувати різницю між вихідними значеннями звичайної та донавченої моделей.

Постає питання: "А де ж виграш?» Адже розміри матриць *W* і *ΔW* мають бути однаковими, отже, у них однакова кількість параметрів, що навчаються, і жодного виграшу в цьому немає." Це дійсно так, якщо ми просто додаємо *ΔW* до *W*. Однак, ми не зобов'язані робити це саме так.

Ось тут і розкривається частина назви - Low Rank. Уявіть собі матрицю, що можна представити як результат перемноження двох менших матриць. Наприклад, матриця розміром 100x70 може мати ранг (кількість лінійно незалежних рядків або стовпчиків) 4 або 20. Це означає, що хоча матриця має 70 стовпчиків, лише 4 (або 20) з них містять суттєву інформацію, а інші просто повторюють або комбінують її.

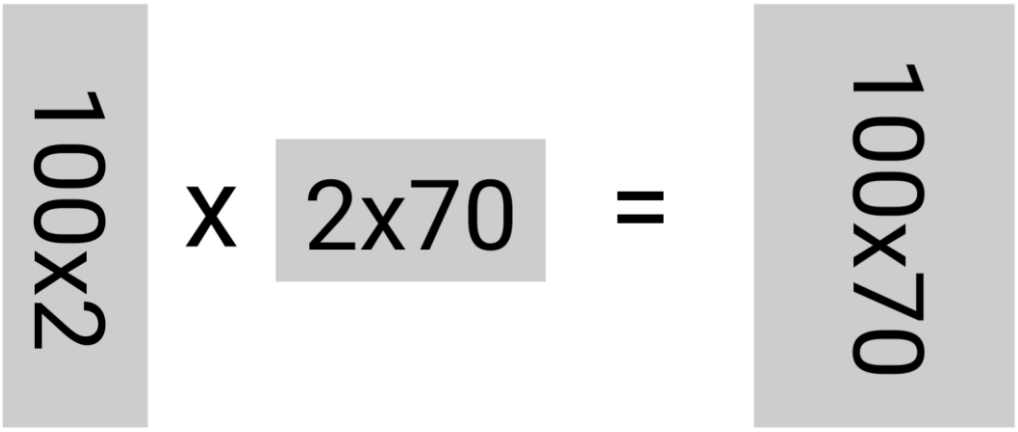


Рис 1.12 Приклад для розмірність вхідного х вихідного простору 100 х 70

Матрицю *ΔW* можна представити як добуток двох інших матриць *A* і *B* **(**рис 1.12**).** Це дозволяє значно зменшити кількість параметрів, що потребують навчання.

Наприклад, на зображенні матриця 100x70 містить 7000 чисел, тоді як дві матриці в лівій частині нерівності містять 140+200=340 чисел. Загалом, за допомогою цього підходу потрібно навчати на *2r/n* менше параметрів, де ***r*** - це

розмір матриць *A* і *B*, який зазвичай обирається невеликим (2-8). Це робить кінцеву кількість параметрів дуже маленькою (приблизно 10^-2).

На жаль, при цьому трохи втрачається загальність, адже ми автоматично припускаємо, що матриця ΔW має низький ранг.

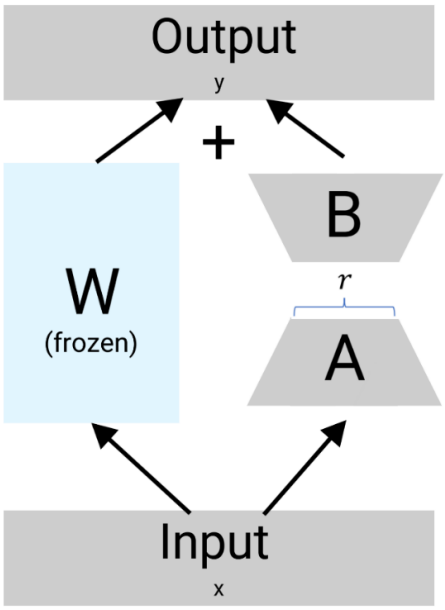


Рис. 1.13 Схема процесу розбиття матриці на менші матриці *A* і *B*

Хоча багато параметрів у великих мовних моделях виглядають "непрацюючими", розробники LoRA з Microsoft запевняють, що це не є проблемою. Вони стверджують, що "внутрішня розмірність" (intrinsic rank) цих моделей дуже низька. Це означає, що більшість параметрів не несуть значущої інформації.

LoRA вирішує цю проблему за допомогою двох матриць: *A* та *B* (рис. 1.13). Вони мають значно меншу розмірність, ніж матриця ΔW вихідної моделі. LoRA навчається добувати "істотні" параметри з *W* за допомогою *A* та *B*, ефективно зменшуючи кількість параметрів, які потребують навчання.

Це було пояснення підкапотного, яка би мовити, процесу роботи LoRA.

Перед тим, як перейти до наступного підрозділу, де ми поговоримо про переваги та недоліки LoRA – підсумуємо і дамо коротке та точне визначення LoRA.

LoRA - це потужний метод тонкого налаштування великих мовних моделей та моделей генерації зображень, який пропонує значні переваги в швидкості, гнучкості та ефективності.

1.3 Переваги та недоліки LoRA-моделей

В цьому підрозділі відокремимо основні переваги та недоліки LoRA від теоретичних матеріалів.

Почнемо з переваг:

а) Значно дешевше донавчання:
   1) навчання моделі для специфічних завдань, використовуючи лише невеликі ресурси;
   2) доступність для широкого кола користувачів, включаючи тих, хто не має потужних комп'ютерів або дорогої підписки на хмарні сервіси;
   3) можливість донавчити модель LoRA на своєму телефоні або Google Colab.

б) Економія місця на диску:
   1) займають значно менше місця на диску, порівняно з традиційними мовними моделями;
   2) простіший процес зберігання та розповсюдження. Наприклад, для моделі GPT-3 з 350 ГБ матриці A та B для всіх лінійних шарів займають лише 35 МБ;

в) Відсутність затримок при висновках:
   1) не впливає на швидкість роботи LLM моделей;
   2) можливість попереднього розрахунку нової матриці W', для уникнення затримок під час використання моделі;

г) Динамічна зміна стилю:
   1) дозволяє динамічно змінювати стиль або поведінку моделі «на льоту»;
   2) можливість уточнення, який стиль подобається користувачеві, з миттєвим оновленням моделі.

Перейдемо до недоліків:

а) Складність реалізації:

1) ґрунтується на складних математичних концепціях;

2) ускладнення реалізації та інтеграції LoRA в існуючі системи машинного навчання.

б) Залежність від базової моделі:

1) спирається на попередньо навчену базову модель, таку як LLaMA, GPT-3, або Stable Diffusion;

2) залежність якості LoRA-моделі від якості базової моделі.

в) Потенційна втрата точності:

1) зниження точності, якщо матриці A та B не підібрані вірно;

2) проблема обробки складних завдань.

г) Обмежена гнучкість:

1) найкраще підходить для лінійних шарів нейронних мереж;

2) застосування до інших типів шарів, таких як рекурентні або згорткові шари, може бути складним або неможливим.

д) Недостатня прозорість:

1) не зовсім зрозумілі внутрішні процеси;

2) ускладнення інтерпретації результатів моделі та діагностики можливих проблем.

LoRA пропонує низку суттєвих переваг, які роблять її перспективним методом для покращення доступності та ефективності великих мовних моделей. Незважаючи на те, що вона все ще перебуває на стадії активного дослідження, існуючі недоліки, ймовірно, будуть вирішені в майбутньому.

Цей розділ був присвячений методам генерації зображень, а також опису LoRA та її призначення. Набуті знання стануть основою для наступного розділу, де ми розглянемо:

— середовище для генерації зображень;

— середовище навчання LoRA-моделей;

— вибір архітектури навчання;

— підбір та підготовку даних;

— процес навчання.

2 РОЗРОБКА LoRA МОДЕЛІ

2.1 Опис середовищ

У цьому підрозділі ми зупинимось на веб-інтерфейсі програми AUTOMATIC1111 Stable Diffusion та kohya-ss.

AUTOMATIC1111 Stable Diffusion, або скорочено - A1111, отримав назву від свого розробника, який підписаний на github як A1111. Це веб-інтерфейс користувача з відкритим вихідним кодом для Stable Diffusion, потужної моделі штучного інтелекту, дає змогу створювати зображення за текстовими описами або модифікувати наявні зображення за допомогою текстових підказок.

Одна з частин в її основі, написана мовою програмування Python, використовує бібліотеки TensorFlow і PyTorch для машинного навчання та обробки зображень. Однак, сам інтерфейс A1111, написаний на HTML, CSS і JavaScript, для створення інтерактивного інтерфейсу, використовує фреймворк React.

Вигляд веб-інтерфейсу A1111 (рис 2.1):

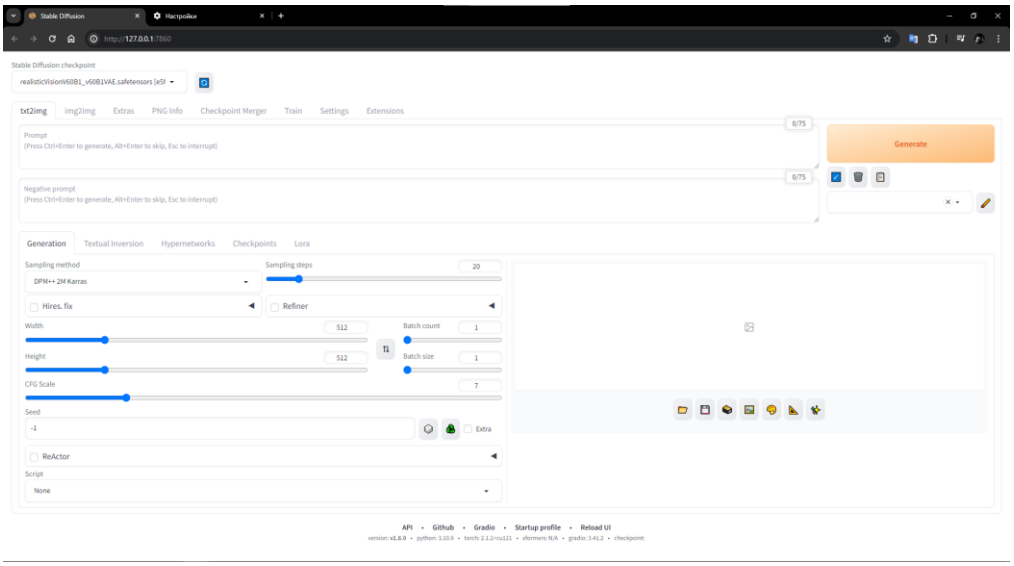


Рис 2.1 Вигляд веб-інтерфейсу A1111

Після того, як розберемось з «консоллю програми», детально оглянемемо кожен важливий елемент інтерфейсу, що будемо використовувати, та визначимо, за що відповідає той чи інший елемент.

Вигляд програми, що відкривається у консольному вигляді (рис 2.2):

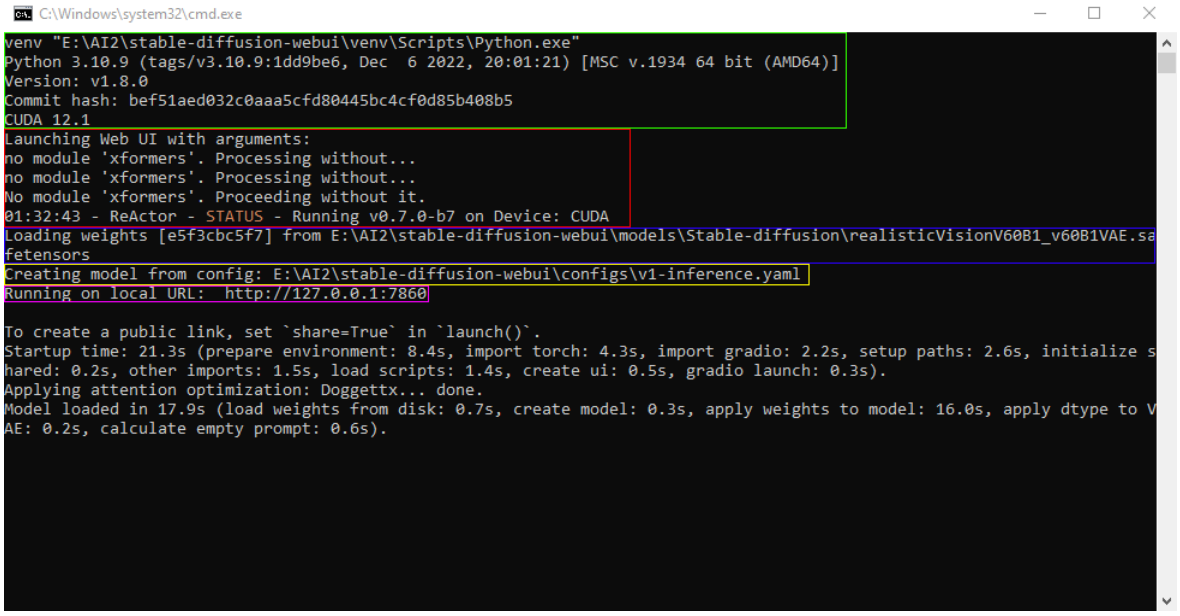


Рис 2.2 Консольний вигляд A1111

Зображення цієї консолі – перше, що побачить користувач перед собою. В процесі роботи з веб-інтрефейсом вигляд консолі буде змінюватися. Коли це буде виникати, я буду показувати відповідну частину консолі, та визнчати, що на ній зображено. На даний момент представлено в різні частини консолі різними кольрами.

— зеленим кольром відображаний шлях до виконавчого файлу Python, верісія A1111. Хеш, який, використовується для того, щоб стягати оновлення для A1111 з github та версія CUDA драйверів, що встановлені на комп’ютері користуача;

— червоним кольром відображені різні модулі (розширення), встановлені в A1111. Якщо модулі встановлені, то під них виділяєтья місце в веб-інтерфейсі;

— синім відображано яка саме Stable Diffusion модель завантажується;

— жовтим кольором відображано шлях, звідки завантажується конфігурація веб-інтерфейсу;

— фіолетовим відображене локальне посилання, що веде до веб-інтерфейсу програми A1111.

Тепер, коли ми ознайомилися з консолюю, можна перейти до веб-інтерфейсу. Маю на меті пояснити, за що відповідає кожен з елемент інтерфейсу, але деякі елементи не буду згадувати через те, що вони не відносяться до нашої задачі.

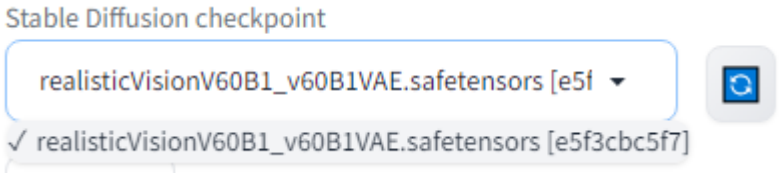


Рис. 2.3 Чекпоінт Stable Diffusion

Чекпоінт Stable Diffusion (рис. 2.3) відповідає за те, яка саме модель буде генерувати зображення. Їх може бути безліч (у нашому випадку - дві) та їх можна змінювати “на льоту” .

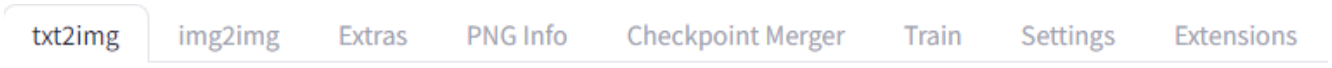


Рис. 2.4 Меню A1111

Меню A1111 (рис. 2.4) представляє собою багато різних вкладинок, між якими можна перемикатися. У вкладинці txt2img ми генеруємо зображення за допомогою текстових запитів. У img2img ми генеруємо зображення з зображення. У Extras можно підвищити роздільну здатність згенерованого зображення. У PNG Info можна подивитися мета-інформацію стосовно згенерованого зображення. Далі

йдуть вкладки, які тим чи іншим чином використовуються для налаштування програми у разі потреби.

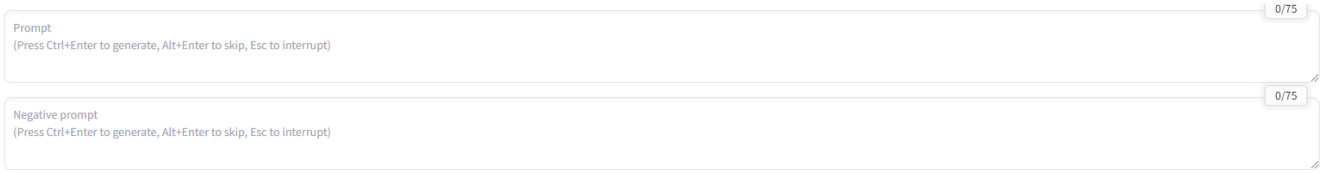


Рис. 2.5 Промти або текстові запити

У цій частині веб-інтерфейсу задаються позитивні та негативні промти (рис. 2.5). Саме через них проходить опис того, що треба зобразити моделі на зображенні, або навпаки чого не повинно бути на зображенні.

Приклад позитивних промтів: selfie, man, smile face, perfect light, best quality. Та негативних: closed eyes, shadow, shadows, dark, dark areas, colored skin, smooth skin. Тобто модель Stable Diffusion, опираючись на ці текстові запити, буде генерувати зображення усміхненого обличчя чоловіка, щоб на ньому не було закритих очей, темних ділянок і т.д.

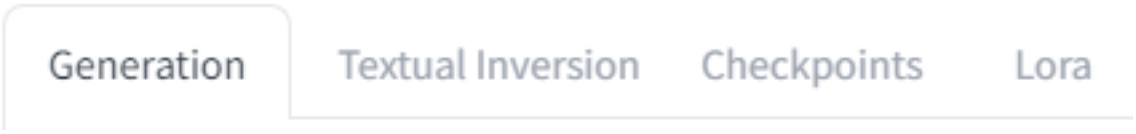


Рис 2.6 Вкладинки налаштувань генерації зображень

У вкладинці Generation (рис 2.6) зняходяться основні налаштування генерації зображень, до яких ми повернемось окремо. У Textual Inversion знаходяться файли, що містять в собі промти, їх можна підключати у поля з промтами. Вкладинка Checkpoints дублює вкладинку згори, у ній можна вибрати яка саме модель буде генерувати зображення. Та вкладинка Lora, у ній знаходяться всі моделі LoRA, що завантажені на ком’пютер та знаходяться у відповідній папці.

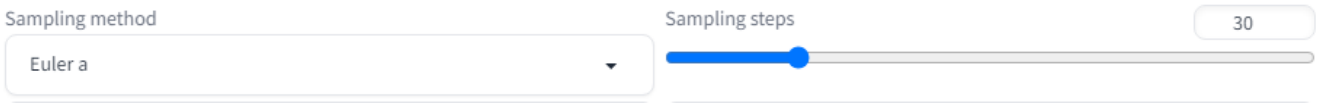


Рис 2.7 Sampling methods і Sampling steps

Методи вибірки (sampling methods) (рис 2.7) – це алгоритми, що використовують для поступового уточнення і деталізації генерованого зображення залежно від сценаріїв використання. Наприклад, коли ми генеруємо фотореалістичне зображення, то ми використовємо метод Euler A, а абстрактне, або мультфільмо-подібне - то DDIM.

Sampling steps задає кількість етерацій (повторів) “денойзингу”, що повинна виконати модель. Видаляючи шум на зображенні, модель Stable Diffusion поступово вимальовує зображення. Цей процес був детальніше описаний у підрозділі 1.1.

Рис 2.8 Налаштувань ширини та висоти

У цій вкладинці (рис 2.8) налаштовується ширина та висота зображення, що генерується.

Рис 2.9 Налаштування CFG Scale

CFG Scale, або Classifier-Free Guidance Scale (рис 2.9), - це параметр, який впливає на те, наскільки суворо згенероване зображення відповідатиме текстовому опису.

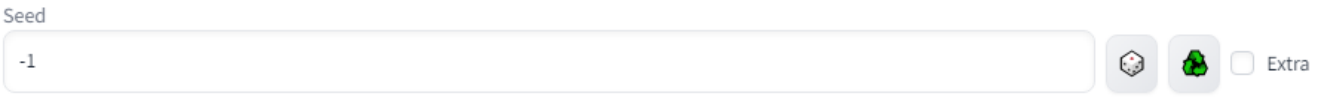


Рис 2.10 Панель Seed

Seed – це панель (рис 2.10), куди можна помістити унікальний номер згенерованого зображення, щоб згенерувати його повторно, але з іншими промтами. Наприклад, фото можуть бути однакові, але на одному зображенні обличчя посміхається, на іншому - ні. Seed “-1” означає, що кожне фото матиме унікальний seed.

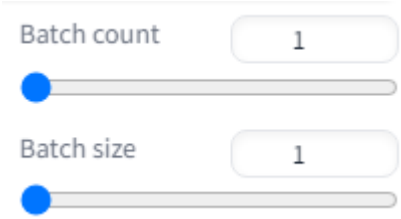


Рис 2.11 Batch count і Batch size

Batch count - визначає загальну кількість зображень, що будуть згенеровані за один запуск моделі. Batch size - визначає кількість зображень, що будуть оброблятися одночасно (рис 2.11).

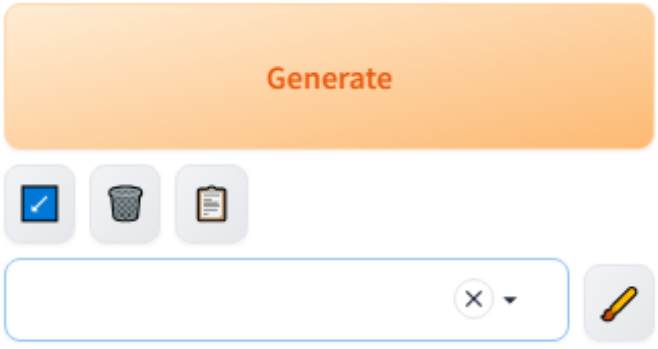


Рис 2.12 Кнопка Generate та спливаюче вікно зі збережженими промтами

Велика кнопка Generate (рис 2.12), запускає «магію» (генерується зображення). Крайня права кнопку, зберігає усі промти, центральна кнопка -

видаляє усі промти з полів для промтів, а крайня ліва кнопка надає можливість перемістити промти з останього зенерованого зображення у поля для промтів. У випадаючому вікні можна вибрати збережені пцромти, щоб перемістити їх у поля для промтів за допомогою кнопки у вигляді пензлика.

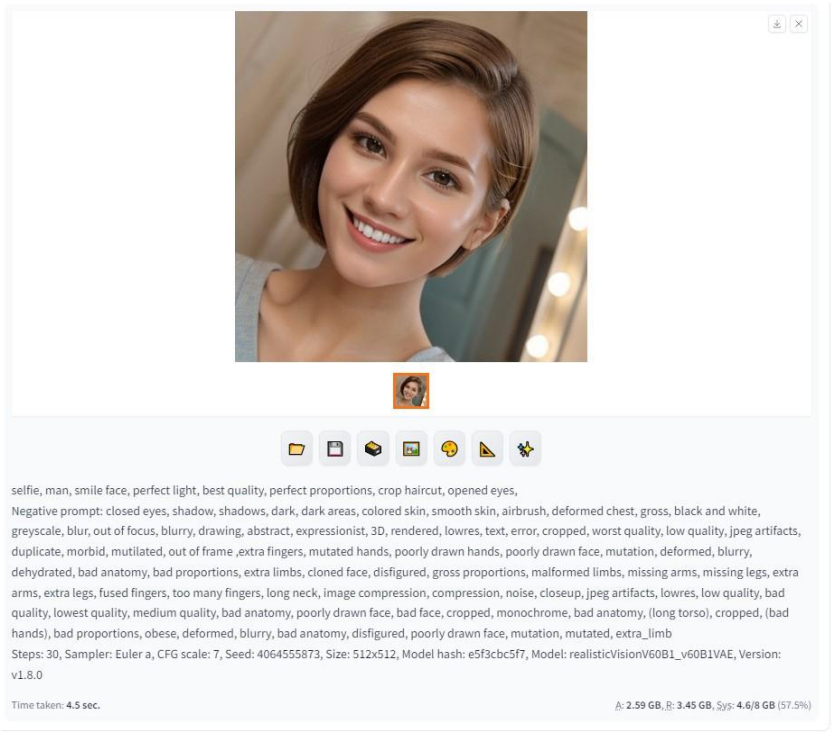


Рис 2.13 Поле з згенерованим зображенням

Останьою частиною інтерфейсу, яку я виділив, є поле, де з’являється згенероване зображення (рис 2.13). Під ним знаходяться позитивні та негативні промти, які були задані. Також є кнопки, які дозволяють перемістити зображення у різні пункти меню.

Рис 2.14 Тамй-лайн генерації зображень

У момент генерації зображень у консолі виводиться тайм-лайн (рис 2.14), де відображається етап процесу генерації зображеня.

Після огляду програми, у якій буде генеруватися зображення, я поверхнево покажу, де проходить процесс навчання LoRA моделі.

Kohya_ss – це програма з відкритим вихідним кодом, призначена для точного налаштування моделей Stable Diffusion та створення LoRA моделей. Вона орієнтована як на початківців, так і на досвідчених користувачів, надаючи зручні інструменти для створення та оптимізації власних моделей ШІ для генерації зображень.

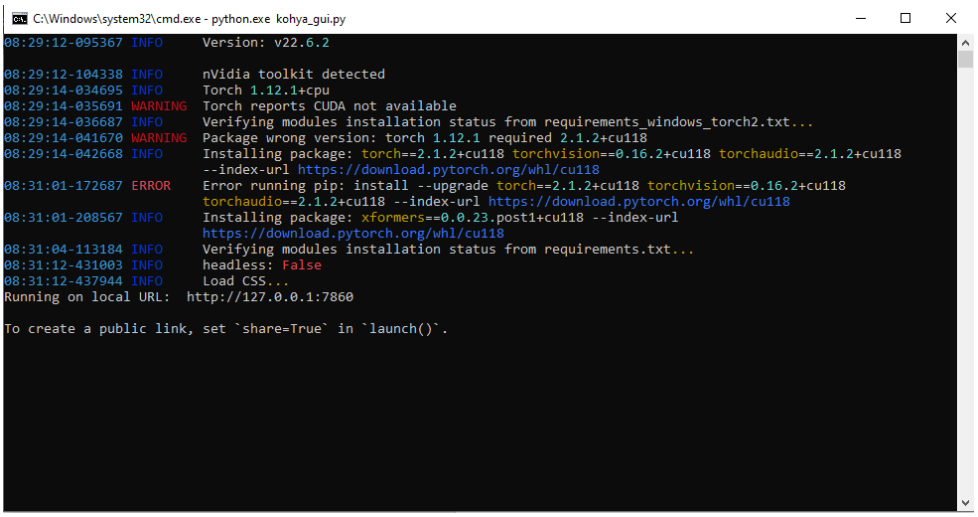


Рис 2.15 Консоль Kohya_ss

Консоль Kohya_ss (рис 2.15) досить схожа на консоль A1111, тому я не буду на ній зупинятися.

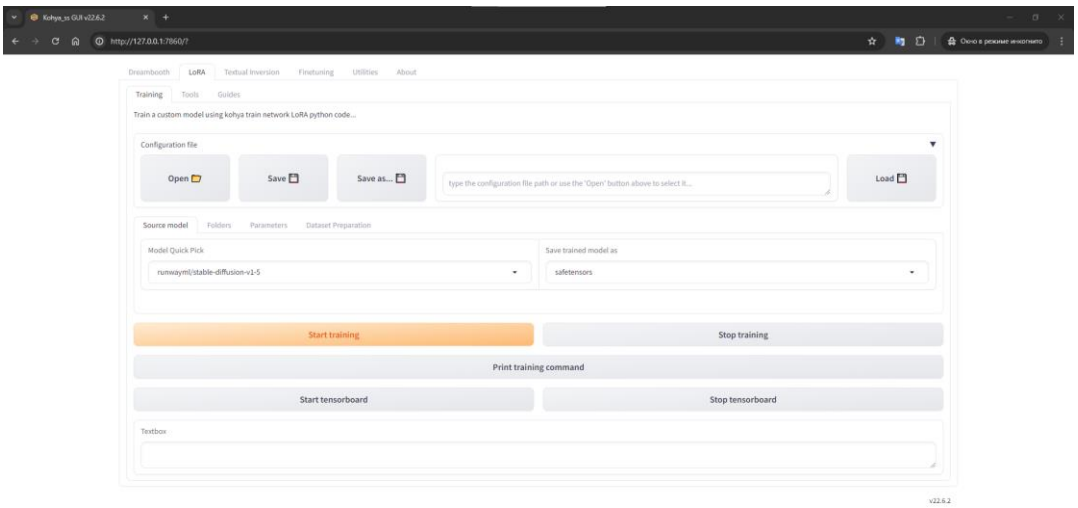


Рис 2.16 Вигляд першої вкладинки веб-інтерфейсу Kohya_ss

Веб-інтерфейс Kohya_ss (рис. 2.16) досить схожий на Stable Diffusion, але розробники у них різні. Перш за все, можна побачити поле, де зберігається та завантажується конфігурація для навчання моделі LoRA. Далі - вкладинка Source model для вибору користувачем бази моделей для навчання LoRA-моделі. Також у цій вкладинці є поле вибору формату збереженої моделі: safetensors, або ckpt. У нижній частині веб-інтерфейсу можна побачити кнопки керування процесом навчання моделі.

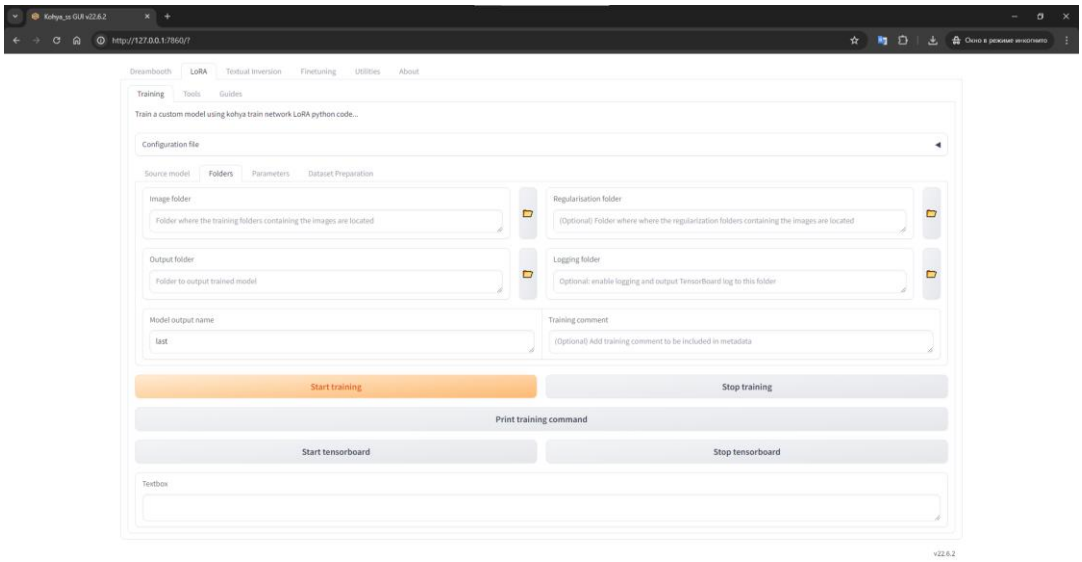


Рис 2.17 Вигляд другої вкладинки веб-інтерфейсу Kohya_ss

На другій вкладинці (рис 2.17) ми бачимо посилання на чотири папки, поле для введення назви моделі та коментар до моделі. Якщо про поле для введення назви і поле для коментарів все зрозуміло, то з чотирма папками все цікавіше. Перша папка під назвою Image folder має містити фотографії для навчання LoRA-моделі. Другу папку під назвою Regularisation folder зазвичай не використовують для LoRA-моделей, але вона слугує для файлів регуляризації. Файли регуляризації використовуються для додавання деяких додаткових обмежень до умови поставленої задачі з метою розв'язати некоректно поставлену задачу або запобігти перенавчанню. Третя папка під назвою Output folder призначена для зберігання новоствореної LoRA-моделі. І остання, четверта папка, використовується для лог-файлів.

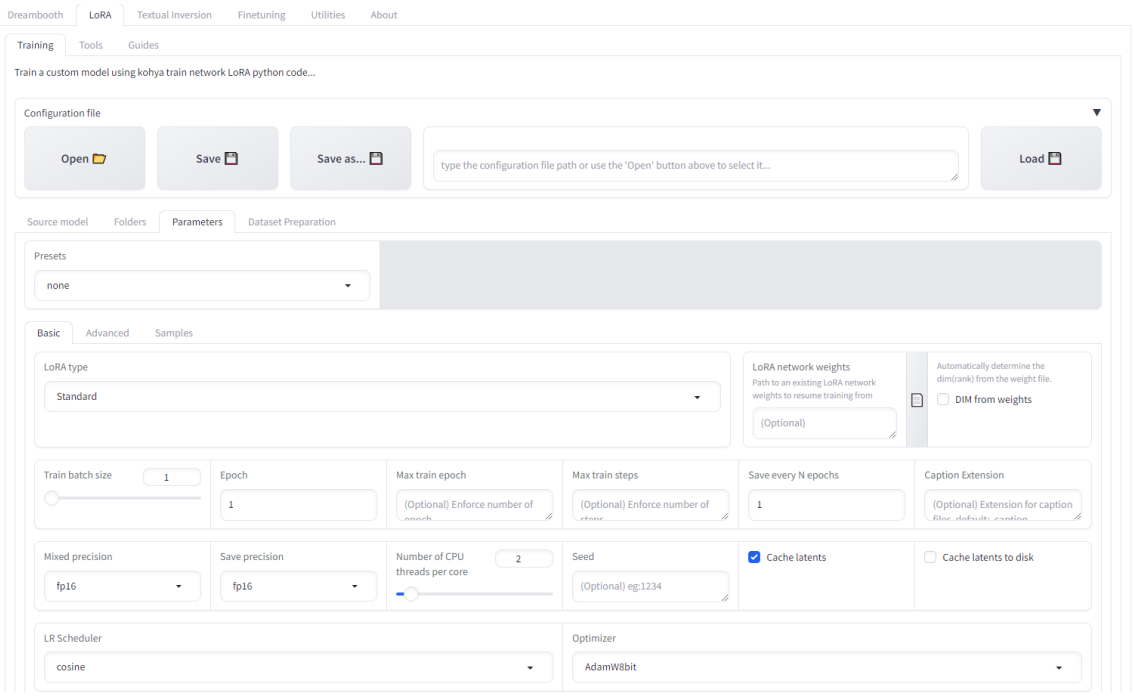


Рис 2.18 Вигляд третьої вкладинки веб-інтерфейсу Kohya_ss

На третій вкладинці (рис 2.18) ми бачимо безліч налаштувань для процесу навчання LoRA-моделі. Ми також торкнемося кожного значущого елементу налаштувань у наступних розділах, після підготовки даних для навчання.

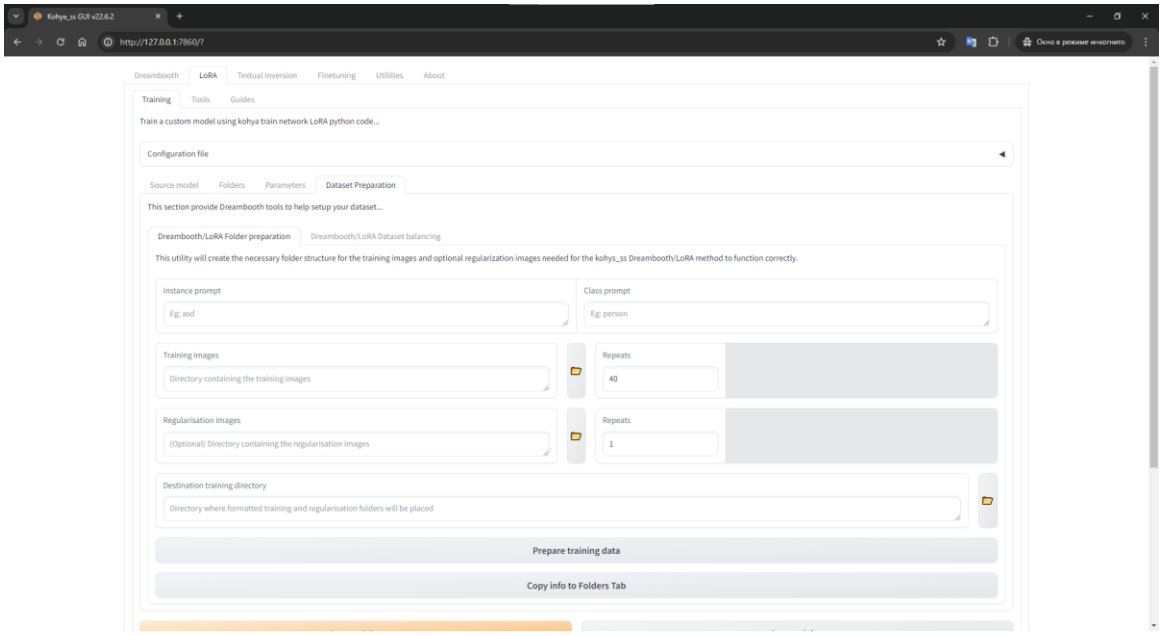


Рис 2.19 Вигляд четвертої вкладинки веб-інтерфейсу Kohya_ss

Ця вкладинка (рис. 2.19) необхідна для створення структур папок для навчальних зображень і додаткових зображень регуляризації, необхідних для коректної роботи методу LoRA.

У підсумку цього підрозділу хотілось би зауважити, що загалом, AUTOMATIC1111 Stable Diffusion - це дуже потужний і універсальний інструмент для створення зображень за допомогою штучного інтелекту, який підходить як для початківців, так і для досвідчених користувачів. А Kohya_ss, у свою чергу, розширює можливості Stable Diffusion, роблячи її доступнішою, ефективнішою, універсальнішою, даючи користувачам можливість тренувати LoRA-моделі.

2.2 Вибір та обґрунтування архітектури

Цей підрозділ знову буде присвячений моделям Stable Diffusion та налаштуванням для процесу навчання LoRA-моделі. На мою думку, моделі Stable Diffusion та налаштування для процесу навчання є основою архітектури, т.я. якість LoRA-моделі залежить орієнтовно на п’ятдесят відсотків. Решта п’ятдесят відсотків це – датасет, але він є предметом іншої підтеми.

Розглянемо моделі. Я багато разів згадував тему великих моделей Stable Diffusion. Вони нам знадобляться для генерації самого зображення та для навчання самої LoRA-моделі. Але де ж її взяти? У попередньому підрозділі ми звернули увагу на консоль, де було вказано, яка саме модель завантажується при старті A1111. Наш приклад - завантажена базова модель Stable Diffusion під назвою “v1-5-pruned-emaonly”, що створена півтора роки тому, коли генерація зображень на десктопах тільки зароджувалась. Вона вже є морально застарілою.

Рис. 2.20 Згенероване зображення моделью v1-5-pruned-emaonly

Це фото (рис. 2.20) було згенероване на текстовий запит “гарний чоловік”. Зображення має досить багато «артефактів». Наприклад: ліве вухо, очі, частина зачіски. І саме зображеняня має не дуже хорошу деталізацію, хоча була задана значна кількість ітерацій генерації.

Що робити в такій ситуації? Тренувати свою модел Stable Diffusion?

Можна створити свою модель, але це складно, вимагає навичок програмування, теоретичних знань по роботі ШІ та дуже великої обчислювальної потужністі комп'ютера. Також, можна навчити модел Stable Diffusion у khoya_ss, але проблему з потребою у великої обчислювальній потужністі комп'ютера це не вирішить. Замість цього я пропоную вибрати одну з моделей представлених на веб-сайті Civitai.com.

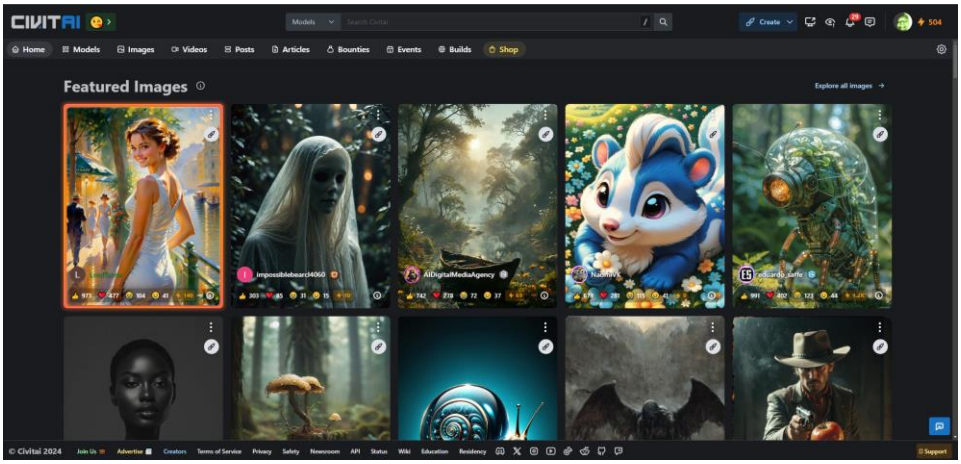


Рис. 2.21 Веб-сторінаа civitai.com

Civitai.com (рис. 2.21) надає платформу для публікації генеративних моделей штучного інтелекту, які можна використовувати для створення зображень, відео та інших творчих матеріалів. Всі моделі, що завантажуються з цього сайту можна безкоштовно використовувати для генерації зображень.

З великого списку моделей, я вибрав одну, яка генерує зображення набагато кращі зображення з тих самих запитів ніж стандартна модель v1-5-pruned-emaonly. Назва вибраної мною моделі для генерації зображень та навчання LoRA-моделі – Realistic Vision.

Рис. 2.22 Згенероване зображення моделью Realistic Vision.

Це зображення (рис. 2.22) також згенероване за текстовим запитом: “гарний чоловік”. Неозброєним оком можна побачити різницю. Майже відсутні артефакти, деталізація і якість в цілому. Civitai.com представлена велика кількість подібних моделей, але саме ця є моїм фаворитом. Надалі вона буде використовуватися для генерації зображень та навчання LoRA-моделі.

У вас знову може з’явитися питання: “А нащо точно налаштовувати цю модель, якщо зображення і так дуже непогано виглядає?” Я буду навчати та використовувати LoRA-модель для того, щоб згенерувати своє обличчя. На жаль, я не можу задати такий текстовий запит моделі Realistic Vision, щоб вона згенерувала

моє обличчя, але якщо я точно налаштую Realistic Vision, використовуючи LoRA-модель, то я уе зроблю. Сподіваюся, це гарний приклад використання LoRA.

Після визначення основної моделі для генерації зображень – повернемося до khoya_ss, саме до налаштувань для навчання LoRA-моделі. Налаштуваня для навчання – важлива складова LoRA-моделі. Від налаштувань залежить не тільки якість LoRA-моделі, а чи буде новостворена модель взагалі працювати.

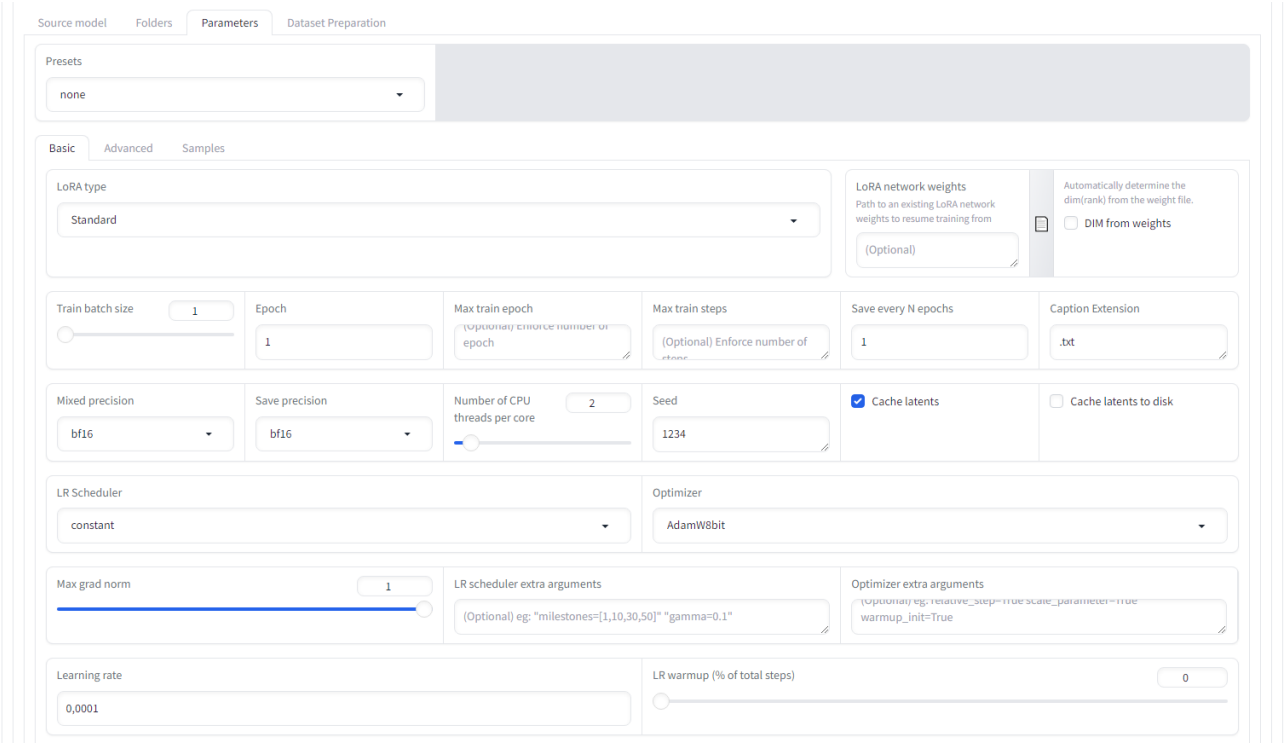


Рис. 2.23 Вкладинка параметрів веб-інтерфейсу Kohya_ss

Kohya_ss надає велику кількість параметрів для налаштувань. З першого погляду (рис. 2.23) можна подумати, що це надає велику гнучкість, але це не зовсім так. Деякі параметри Kohya_ss, на жаль, працюють некоректно. З цього приводу, буде раціонально звернути увагу на самі значущі з них, та надати їх опис, для кращого розуміння створення LoRA-моделі на практиці.

Epoch – це серія усіх процедур та алгоритмів, які вивчають зображення. Зазвичай одна epoch дорівнює одній сотні ітерацій навчання на одному зображенні. Цікаво зазначити, що кількість ітерацій можна змінювати, задаючи їх у вигляді назви для папки, у якій знаходяться зображення для навчання. На практиці, цей

параметр завжди слід залишати на значенні один, тому що epoch це комплекс всіх заходів, який повинен бути проведений тільки один раз.

Train batch size – кількість паралельних пакетних операцій обробки зображення для навчання. Наприклад: відеокарта буде оброблювати одне зображення, якщо параметр train batch size дорівнює одному, якщо - двом, відеокарта буде одночасно обробляти два зображення, і т.д. На практиці, це зменшує час навчання моделі, тому що за одну epoch (тобто за сто ітерацій) відеокатра оброблює два зображення для навачання. З цього можна зробити висновок, що це у гіршу сторону впливає на модель. Цей параметер варто залишати на значенні один.

Caption Extension – поле у якому вводить назва розширення файлу. Файли заданого типу, мають містити текстовий опис до зображень, на базі яких буде вчитися модель. У процесі навчання модель буде зчитувати опис відповідного файлу та зіставляти його з тим, що знаходиться на зображенні. У моєму випадку розширення файлу – .txt.

Mixed precision – тип алгоритму змішаної почності. Це означає, що різні частини моделі можуть використовувати різні формати даних для оптимізації балансу між продуктивністю та точністю. FP16 використовує 16 біт для представлення чисел, що вдвічі менше, ніж 32-бітний формат FP32, що зазвичай використовують у машинному навчанні. Це дає змогу збільшити швидкість обчислень і скоротити споживання пам'яті, що є великою перевагою для користувачів з обмеженими ресурсами. BF16, в свою чергу, використовує ще більш компактний формат – 8 біт + 8 біт експоненти, що ще більше прискорює обчислення і знижує вимоги до пам'яті, хоча і з деяким компромісом у точності. На практиці, вибір цього параметру залежить від відеокарти. BF16 обчислюється на тензорних ядрах відеокарт Nvidia серії Ampere. Якщо використовувати BF16 з відеокартами, які не мають тензорних ядер, то kohya_ss не запуститься. Я буду використовувати BF16, так як маю відеокарту Nvidia серії Ampere.

Cache latents - це функція, що дозволяє використовувати центральний процесор для прискорення обробки зображень за рахунок кешування проміжних даних. У налаштуваннях цей параметр має бути увімкнений.

Learning rate – значення, яке впливає на швидкість навчання моделі. Чим нижче значення – тим повільніше вона навчається. Цей параметр, залишаємо у страндартному значенні 0,0001.

LR Scheduler - це інструмент, який дозволяє вам змінювати швидкість навчання (learning rate) вашої моделі під час процесу навчання. Це може бути корисно для поліпшення продуктивності та досягнення більш якісних результатів. Існує кілька різних типів LR Schedulers, кожен з яких має свої переваги та недоліки. Деякі популярні варіанти включають:

— Constant: зберігає швидкість навчання постійною протягом усього навчання;
— Cosine Annealing: знижує швидкість навчання в міру наближення до завершення навчання;
— Polynomial Decay: знижує швидкість навчання з використанням поліноміальної функції;
— Adafactor: автоматично регулює швидкість навчання на основі градієнтів.

Вибір LR Scheduler залежатиме від ваших конкретних потреб і цілей навчання. Я залишу Constant за замовчування, т.я. у деяких випадках інші типи LR Schedulers можуть призвести до погіршення якості моделі.

Network Rank (Dimension) – кількість аспектів, які будуть враховані при навчанні, все що нейронна мережа зможе визначити на фото і буде аспектом. Наприклад: зачіска, колір очей, формма обличча і т.д.. Цей параметр не слід встановлювати вище 200, коли збільшується ризик перенавчання. Залишимо цей параметр на відмітці 128.

Max resolution – максимальне разширення на фото, на якому буде навчатися модель. В-подальшому, всі зображення датасету будуть даунскейлитись до цього значення.

Це були основні налаштування Kohya_ss для навчання LoRA моделі. Інші параметри, можна залишити за замовчуванням, якісь з них, маючи оптимальне значення, працювати не будуть.

2.3 Підготовка даних

Зупинимось на даних, які плануємо використовувати для навчання LoRA-моделі, та як їх слід пдготувати для навчання. Як я зазначав раніше, маю на меті використати зображення свого обличча для навчання LoRA-моделі. На мою думку – це досить цікава ідея, яка може наглядно продемонстувати можливості LoRA.

Почнемо з критеріїв підбору даних. Проходячи преддипломну практику у компанії Huawei Україна, я пройшов курс зі штучного інтелекту, де розповідалось про необхідні критерії та вимоги для створення якісного дата-сету:

а) Повнота даних:
   1) відсутність пропусків: всі важливі метадані (такі як дата і час зйомки, роздільна здатність, місце зйомки) повинні бути заповнені, а пропуски мінімальні або оброблені;
   2) достатній обсяг даних: датасет має бути достатньо великим, щоб представляти всі можливі сценарії і варіанти для задачі;

б) Коректність даних:
   1) точність: фотографії повинні бути чіткими і правильний ракурс, без помилок, таких як розмиття або неправильна експозиція;
   2) актуальність: фотографії повинні бути актуальними та відповідати поточному часу, якщо це необхідно для задачі.

в) Консистентність даних:
   1) єдиний формат: Всі фотографії повинні бути в одному форматі файлів (наприклад, всі у JPEG або PNG);

2) однорідність: фотографії повинні мати однаковий стиль, освітлення і роздільну здатність, якщо це можливо;
3) розмірність: всі фотографії повинні мати однакове розширення. Наприклад: 512x512, або 768x768 пікселів.

г) Повнота і адекватність атрибутів:
1) релевантність: метадані та додаткові інформаційні поля (такі як теги чи категорії) повинні бути релевантними для задачі і мати сенс у контексті аналізу;
2) наявність ключових фіч: Датасет повинен містити всі важливі для задачі атрибути, які можуть значно впливати на результат;

д) Відсутність шуму і артефактів:
1) очистка даних: фотографії повинні бути очищені від шуму, артефактів і дефектів, що можуть негативно впливати на аналіз;
2) обробка артефактів: артефакти, такі як зображення з непотрібними об'єктами, повинні бути ідентифіковані і, по можливості, видалені або оброблені.

е) Документованість і доступність метаданих:
1) опис даних: датасет має супроводжуватись докладним описом полів (метаданих), їх значень, допустимих діапазонів і форматів;
2) джерело і ліцензія: інформація про походження фотографій і умови їх використання повинні бути чітко вказані.

ж) Відсутність дублікатів:
1) унікальність зображень: датасет має бути перевірений на наявність дублікатів, і дублікатні фотографії повинні бути видалені.

з) Безпека даних:
1) анонімізація: фотографії, що містять особисту інформацію, повинні бути анонімізовані для дотримання конфіденційності і захисту даних;
2) захист даних: датасет повинен відповідати стандартам безпеки і конфіденційності, особливо якщо дані чутливі або особисті.

На жаль, мій датасет не зможе відповідати всім вищезазначеним критеріям. Датасет не відповідає таким категоріям: безпека даних, документованість і доступність метаданих. Оскільки я використовую зображення свого обличчя для навчання LoRA-моделі, не можу провести анонімізацію даних. З тієї ж причини фотосет не має чіткої ліцензії на використання.

Я не вважаю, що порушення цих критеріїв негативно вплине на новостворену модель. Важливо зазначити, що ця модель не буде у відкритому доступі, тому можна порушити деякі критерії, що стосуються копірайту та конфіденційності даних для навчання. Однак, не можна забувати про дотримання копірайту та конфіденційності у випадку, коли модель буде у подальшому відкритому доступі.

Наприклад, Getty Images подала до суду на Stability AI, компанію, яка розробила штучний інтелект, що генерує стабільні дифузійні зображення. За словами компанії, Stability AI побудувала конкурентний бізнес на несанкціонованому використанні понад 12 мільйонів зображень з фотобанку Getty без виплати компенсації. На думку Getty, це "нахабне порушення авторських прав у приголомшливих масштабах".

Тобто, на практиці, головними критеріями до датасету є відсутність артефактів, однорідність, розмірність, точність, однаковий формат. Після відповідних маніпуляцій, по підвищенню якості датесету, він має такий вигляд:

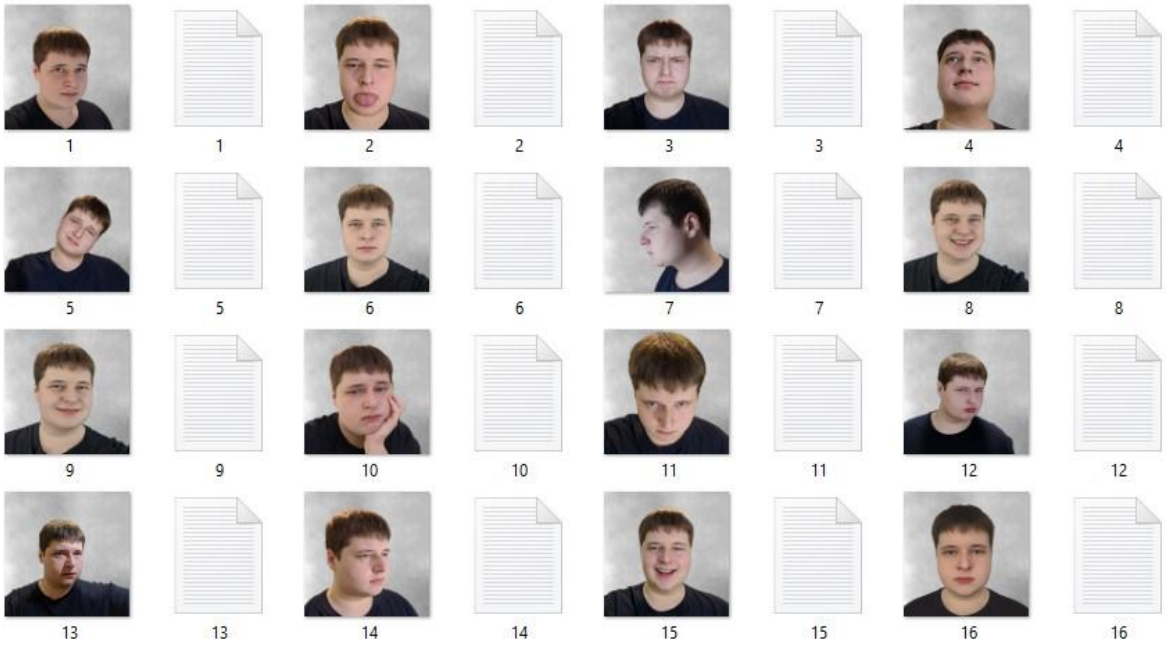


Рис. 2.24 Вигляд датасету з зображеннями та текстовим описом до них

Датасет складається з 16 зображень та 16 текстових описів до них (рис. 2.24). Назви зображень відповідають назвам текстових файлів з їх описами. Зображення в датасеті показують моє обличчя з різних ракурсів, при різному освітленні та з різними емоціями. Це забезпечує різноманітність даних, що дозволяє моделі краще навчитися та запобігає перенавчанню. 16 зображень – це мінімально необхідна кількість для навчання LoRA моделі; якщо їх буде менше, модель може стати "упередженою". Фон на зображеннях змінений, щоб сфокусувати навчання на обличчі. Усі фото мають однаковий розмір – 512x512 пікселів.

Текстовий опис знаходиться у файлах розширення .txt. Наприклад, текстовий опис для зображення 2 має такий вигляд:

“Vadym Panchenko, man, simple background, a man sticking his tongue out with a black shirt”.

У процесі навчання LoRA-моделі, модель буде зіставляти текстовий опис з тим, що знаходиться на зоюраженні.

2.4 Навчання та оптимізація моделі

У цьому підрозділі ми налаштуємо параметри kohya_ss, описані в попередніх розділах, створимо необхідну структуру папок для навчання та розпочнемо процес навчання моделі LoRA.

Почнемо зі структури папок:

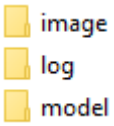


Рис. 2.25 Вигляд структури папок для навчання моделі LoRA

У папці image (рис. 2.25) знаходиться інша папка під назвою 100_VadymPanchenko. У цій назві 100 – кількість ітерацій, яка рівняється одній epoch, а VadymPanchenko – ключове слово, по якому буде визиватися LoRA-модель. Процес визову LoRA-моделі по ключовому слову, буде описаний у наступному розділі.

У папці log містяться log-файли. Log-файли в khoya_ss потрібні для моніторингу процесу, відстеження помилок, аналізу продуктивності, аудиту, дебагінгу та забезпечення зворотного зв'язку, допомагаючи підтримувати прозорість, ефективність та стабільність системи.

У папці model з’явиться новостоврена LoRA-модель. У моєму випадку вона матиме назву “VadymPanchenko_v1”, з розширенням .safetensors.

Перейдемо до налаштувань, які ми обговорили у попередній підтемі. Повторювати їх немає сенсу, тому я просто покажу заповнені поля з налаштуваннями.

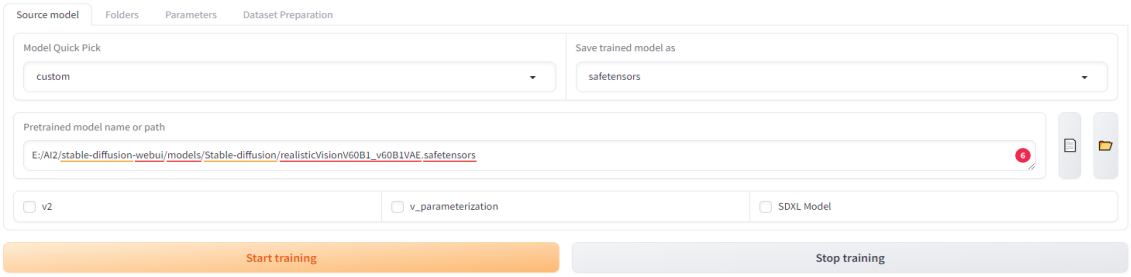


Рис. 2.26 Паналь налаштувань Source model

У панелі source model (рис 2.26) я задав місце знаходження великої Stable Deffusion моделі, та розширення в якому буде збережена новостворена LoRA-модель.

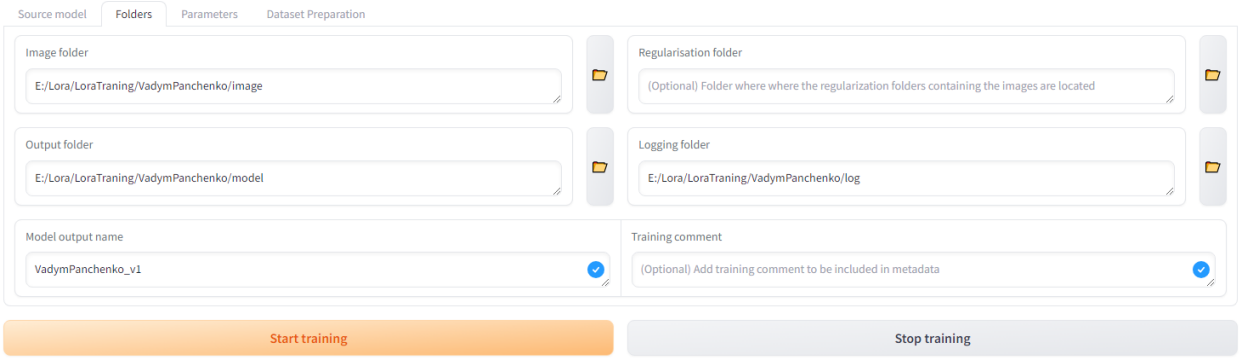


Рис. 2.27 Панель налаштувань Folders

У панелі folders (рис 2.27) я задав місце знаходження папок image, log, model. Їх призначення ми вже обговорювали. Та ввів назву моделі, яку буду створено.

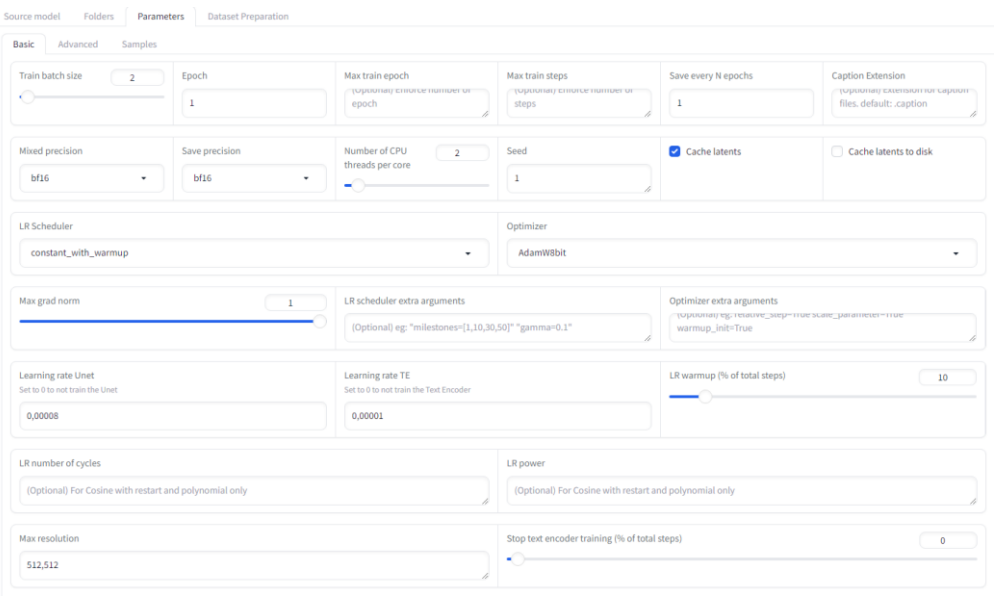


Рис. 2.28 Панель налаштувань Parameters

У панелі налаштувань Parameters (рис 2.28) я встановив параметри, визначені в підрозділі 2.2. Після цього можна натиснути кнопку "Start Training" для початку процесу навчання моделі.

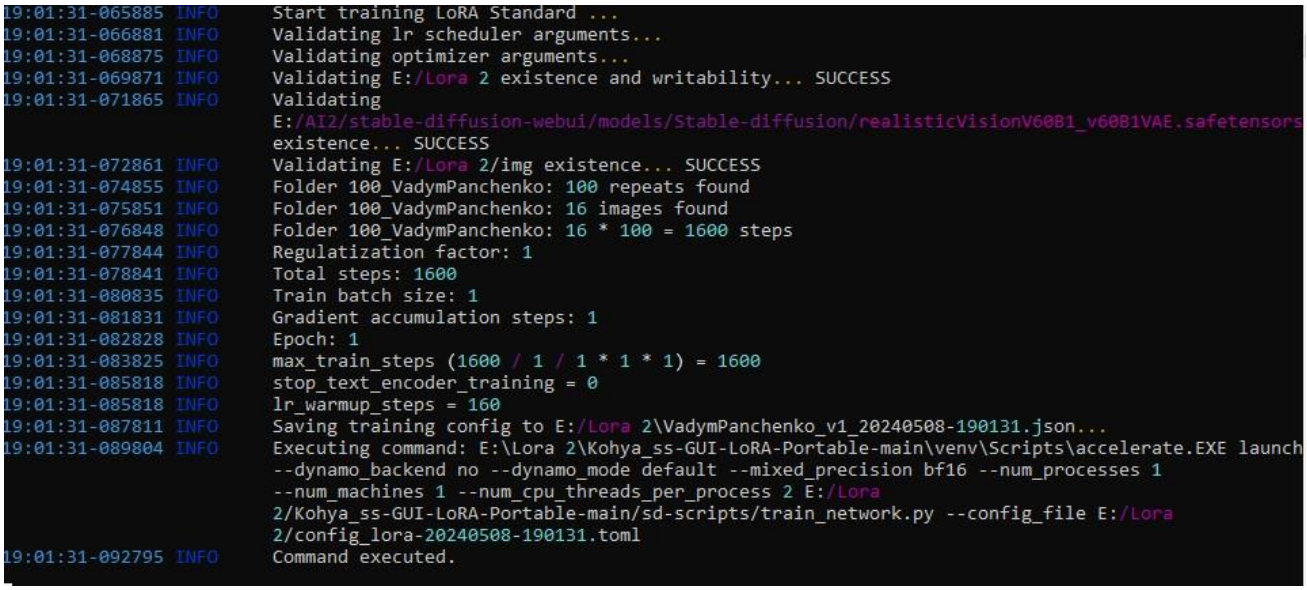


Рис. 2.29 Консоль khoya_ss у момент навчання LoRA-моделі

Давайте докладніше розглянемо, що саме описується в деяких рядках консолі (рис. 2.29):

а) 01:31:31-071865 INFO - Перевірка наявності та доступу до файлу моделі за шляхом E:/AI2/stable-diffusion-webui/models/Stable-diffusion/realisticVisionV60B1_v60B1VAE.safetensors. Перевірка успішна.

б) 01:31:31-072861 INFO - Перевірка наявності зображень у директорії E:/Lora 2/img. Перевірка успішна.

в) 01:31:31-074855 INFO - Знайдено 100 ітерацій у папці 100_VadymPanchenko.

г) 01:31:31-074881 INFO - Знайдено 16 зображень у папці 100_VadymPanchenko.

д) 01:31:31-074885 INFO - Обчислення загальної кількості кроків: 16 зображень * 100 повторень = 1600 кроків.

е) 01:31:31-078884 INFO - Встановлення коефіцієнта регуляризації на 1.

ж) 01:31:31-078881 INFO - Встановлення загальної кількості кроків навчання на 1600.

з) 01:31:31-078888 INFO - Встановлення розміру партії (batch size) на 1.

и) 01:31:31-078889 INFO - Встановлення кількості кроків накопичення градієнтів на

к) 01:31:31-078891 INFO - Встановлення кількості епох на 1.

л) 01:31:31-078892 INFO - Встановлення максимального числа кроків навчання на 1600.

Рис. 2.30 Таймлайн у консоль khoya_ss у момент навчання LoRA-моделі

Після цього розпочинається процес навчання моделі, і в консолі з'являється таймлайн (рис. 2.30). За ним можна визначити приблизний час, необхідний для завершення навчання моделі.

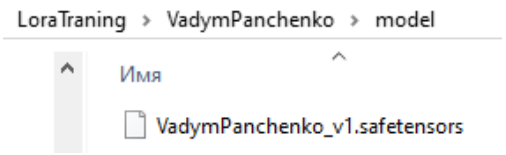


Рис. 2.31 Створена LoRA-модель

Після завершення навчання у вибраній раніше папці з’явиться новостворена LoRA-модель (рис. 2.31). Її назва буде такою ж, яку ми задали. Ця модель готова до використання. Однак, процес її використання буде розглянуто в наступному розділі.

У цьому підрозділі я описав процес підготовки даних та навчання LoRA-моделі на власних фотографіях. Використовуючи 16 зображень свого обличчя з різними емоціями та освітленням, а також текстові описи до кожного зображення, я очистив дані від артефактів і привів їх до однакового розміру. Навчання здійснював за допомогою kohya_ss з параметрами, описаними раніше, у розділі 2.2

3 ТЕСТУВАННЯ ТА ОЦІНКА МОДЕЛІ

3.1 Використання створеної моделі

Щоб використати створену LoRA-модель у A1111, необхідно перемістити файл моделі до відповідної папки A1111, де зберігаються всі LoRA-моделі.

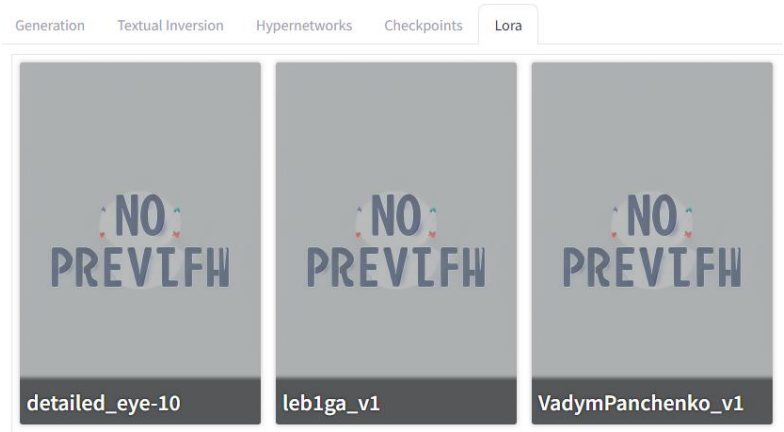


Рис. 3.1 Вкладинка Lora у A1111

Після цього модель буде доступна у вкладці "Lora" в інтерфейсі A1111 (рис. 3.1). При натисненні на LoRA-модель у цій вкладинці вона потрапить у поле з промтами.

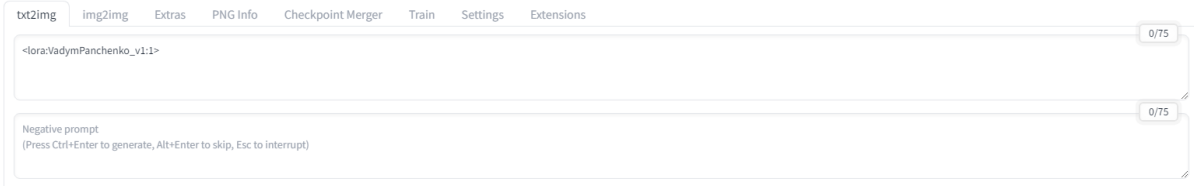


Рис. 3.2 Поле з промтами у A1111

Після того як ми обрали потрібну LoRA-модель, у нашому випадку це VadymPanchenko_v1, у полі з промтами (Рис. 3.2) з’явилася структура <lora:VadymPanchenko_v1:1>, де “lora:” – це тип викликаної моделі, VadymPanchenko_v1 – її назва, а 1 – вага. На практиці вага визначає, наскільки

сильно LoRA-модель впливатиме на зображення. Наприклад, якщо встановити вагу на 0.1 (<lora:VadymPanchenko_v1: 0.1 >), вплив моделі буде слабким, і навпаки, чим більша вага, тим сильніший вплив. Ми ще повернемося до порівняння різних значень ваги на практиці в наступних розділах.

Але для використання LoRA-моделі цього недостатньо. Пам’ятайте у минулому розділі ми створювали папку з назвою 100_VadymPanchenko, де 100 кількість ітерацій навчання моделі на одному фото, а VadymPanchenko, як би мовити, ключ активації, по якому активується LoRA-модель. Цей ключ, при необхідності можна знайти у метаданих моделі, використовуючи A1111 (рис. 3.3).

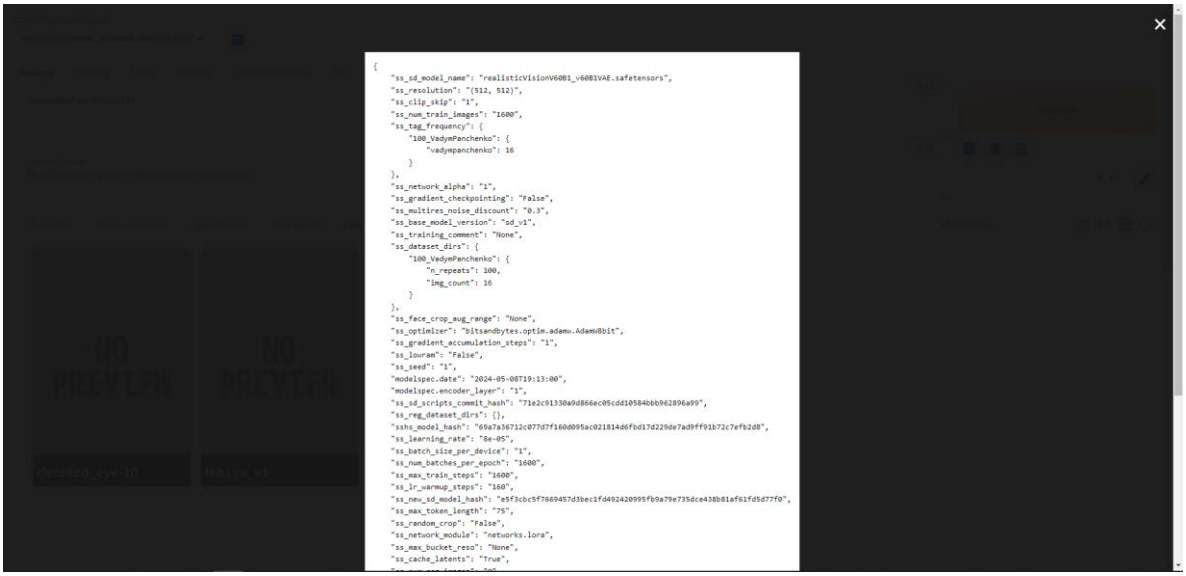


Рис. 3.3 Метадані моделі у A1111

При використанні структури <lora:VadymPanchenko_v1:1>, та ключа VadymPanchenko, можна починати генерувати зображення. Проте для досягнення кращих результатів рекомендується додати додаткові позитивні та негативні промпти з детальним описом того, що саме повинно і не повинно бути зображено.

Промпти, або текстові запити для генерації зображень, обмежені лише вашою уявою та орфографією. У наступних підрозділах ми спробуємо різні комбінації налаштувань генерації та промптів. А зараз я використаю стандартний набір промптів, який дозволить створити якісне зображення.

Після введення промтів та налаштувань генерації зображень можна згенерувати перше зображення.

Рис. 3.4 Згенероване зображення за допомогою створеної LoRA-моделі

Це зображення (рис. 3.4) було згенеровано за такими позитивними текстовимизапитами:

<lora:VadymPanchenko_v1:0.65>, VadymPanchenko, young man, closeup portrait photo man, 8k uhd, high quality, dramatic, cinematic, white background, good skin, smooth skin

Та негативними:

(Beard, bristle, mustache, moles: 1.2, red cheeks), red skin, irritated skin, moles on cheeks: 1.2, brown dots, (deformed iris, deformed pupils, semi-realistic, cgi, 3d, render, sketch, cartoon, drawing, anime), text, cropped, out of frame, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, dehydrated, bad anatomy, bad proportions, extra limbs, cloned face, disfigured, gross proportions, malformed limbs, missing arms, missing legs, extra arms, extra legs, fused fingers, too many fingers, long neck.

Як ви бачите, модель працює успішно. Вона генерує реалістичні зображення, дуже схожі на ті, що були в датасеті. Можна сказати, що завдання дипломної роботи виконане – LoRA-модель була створена. Проте, я пропоную не зупинятися на цьому, а протестувати модель, використовуючи різні налаштування, промпти та інші LoRA-моделі. Також порівняємо згенеровані зображення із зображеннями з датасету.

3.2 Тестування LoRA-моделі з різними налаштуваннями

Розглянемо на практиці, як працюють кроки семплінгу (Sampling steps). Sampling steps – це кількість ітерацій видалення шуму з зображення, тобто кількість повторів, необхідних для генерації зображення методом стабільної дифузії. Я згенерую одне й те саме зображення з різними значеннями Sampling steps, використовуючи однаковий Seed. Зображення будуть розміщені зліва направо, від меншого значення Sampling steps до більшого.

Рис. 3.5 Вплив Sampling steps на якість згенерованого зображення

Можна спостерігати, як зображення стає більш чітким і деталізованим завдяки видаленню шуму (рис. 3.5). Число у верхньому лівому куті кожного зображення відображає кількість ітерацій цього процесу.

Наступним етапом я планую порівняти різні семплери. Для цього я використаю однакові промпти та Seed, змінюючи лише метод семплінгу, щоб побачити різницю між ними. Найпопулярнішим з них є Euler A, і саме його я вибрав у налаштуваннях під час навчання моделі. Проте, на практиці LoRA-модель можна використовувати з будь-яким семплером.

Рис 3.6 Вплив семплерів на зображення, що генерується

— DPM++ 2M Karras: Цей семплер використовує метод з двома моментами (2M) для покращення якості зображень. Він збалансований між швидкістю та якістю, створюючи реалістичні зображення.
— DPM++ SDE Karras: Використовує стохастичне диференціальне рівняння (SDE) для точнішого моделювання зображень. Цей семплер забезпечує стабільні і якісні результати.
— Euler A: Популярний завдяки простоті та ефективності. Використовує метод Ейлера, що забезпечує швидкий і надійний результат, але може поступатися в деталізації та якості.
— DPM++ 2M SDE Karras: Поєднує методи з двома моментами (2M) та SDE, що дозволяє досягти високої якості зображень. Це один з найкращих семплерів для створення складних і детальних зображень.

Отже, на практиці семплер вибирається відповідно до конкретного завдання. На мою думку, всі вони досить ефективні. Однак деякі семплери можуть створювати більш деталізовані зображення, тоді як інші можуть зменшити час, необхідний для генерації (рис 3.6).

Далі, я хочу протестувати цю LoRA-модель у спільному використанні з іншими LoRA-моделями. Їх можна завантажити з сайту civitai.com, про який я розповідав у минулому розділі.

Рис. 3.7 Використання моєї LoRA-моделі з іншою

Я завантажив іншу LoRA-модель з сайту civitai.com, яка дозволяє генерувати зображення в нуарному стилі. Підключивши цю модель та налаштувавши промпти, я отримав нуарні зображення зі своїм обличчям (рис. 3.7).

Загалом, для надання зображенню потрібного стилю можна обійтися без використання інших LoRA-моделей. Ось приклад таких зобажень:

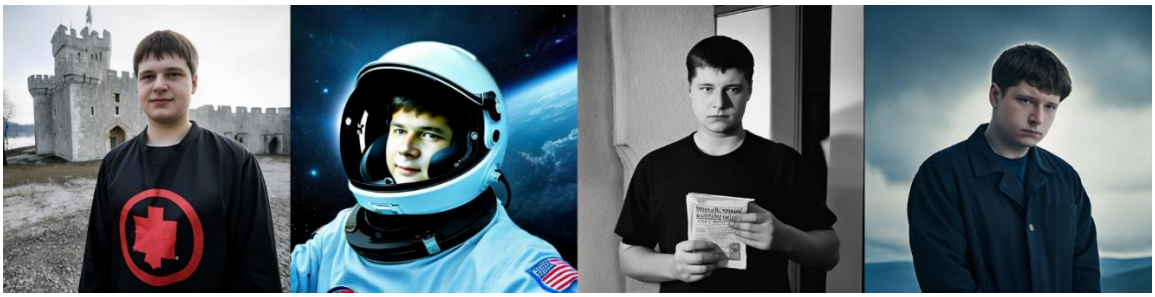


Рис. 3.8 Використання моєї LoRA-моделі

Усі ці зображення (рис. 3.8) згенеровані виключно за допомогою створеної нами моделі, використовуючи різні текстові запити. На першому зображенні зліва я намагався згенерувати середньовічне місто, і, на мою думку, результат вийшов досить непоганим. На другому зображенні я використав промти для створення космічного стилю. Третє зображення було згенеровано за тими ж промтами, що й зображення на рисунку 3.7, але без використання LoRA-моделі, яка додає можливість створювати нуарні зображення. Хоча зображення вийшло непоганим, воно не настільки деталізоване, як ті, що згенеровані з додаванням нуарної LoRA-моделі. Останне зображення – одни з портретів, згенерованих у процесі роботи.

Узагальнено, ця підтема показує, що моделі LoRA, Stable Deffusion та A1111 відкривають широкі можливості для творчого процесу у генерації зображень. Експериментуючи з різними налаштуваннями та комбінуванням різних моделей LoRA, можна досягти бажаного ефекту на зображенні.

3.3 Оцінка якості згенерованих зображень

Наприкінці, для оцінки якості згенерованих зображень, я планую провести порівняння між зображеннями з датасету та тими, що були створені на основі текстових описів, використаних під час тренування LoRA-моделі.

Рис. 3.9 Зображення та текстовий опис зображення

Зокрема, я маю намір використовувати описи зображень як текстові запити для генерації нових зображень (рис 3.9). Після цього планую порівняти згенеровані зображення з оригінальними з датасету. Таким чином, можна буде наочно

порівняти згенеровані зображення з оригінальними, що дозволить суб’єктивно оцінити їх якість.

Для наглядності я порівняю три пари зображень:

Рис. 3.10 Поріняння зображення з датасету та згенерованого LoRA-моделю №1

На лівому зображенні (рис. 3.10) з датасету для навчання LoRA-моделі зображений я з коротким світло-каштановим волоссям і світлою шкірою. Я виглядаю серйозним або трохи сердитим, з нахмуреними бровами і стиснутими губами. Фон білий, нейтральний.

На правому зображенні (рис. 3.10), згенерованому за допомогою LoRA-моделі, чоловік який досить схожий на мене, але з більш агресивним або незадоволеним виразом обличчя. На згенерованому зображенні я також дивиться у камеру з опущеною головою, створюючи відчуття загрози або напруги. Фон сірий, текстурований.

Зображення з датасету виглядає більш природнім, тоді як згенероване – більш вираженим і напруженим.

Рис. 3.11 Поріняння зображення з датасету та згенерованого LoRA-моделю №2

На лівому зображенні (рис. 3.11), з датасету для навчання LoRA-моделі, я дивлюсь вбік, вираз обличчя задумливий або злегка сумний. Я одягнений у темну футболку, а фон сірий і текстурований.

На правому зображенні (рис. 3.11), згенерованому за допомогою LoRA-моделі, я з коротким світло-каштановим волоссям і світлою шкірою. Я дивлюсь вбік, вираз обличчя спокійний і нейтральний. На мені темно-зелена куртка, а фон білий і розмитий, що дозволяє зосередитися на обличчі.

Зображення з датасету виглядає також быльш природнім, тоді як згенероване зображення створює кінематографічний настрій завдяки гарному ракурсу та деталізованому освітленню.

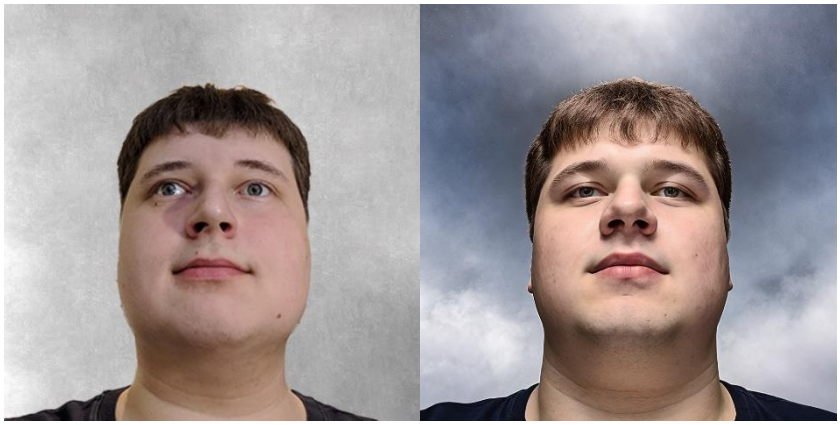


Рис. 3.12 Поріняння зображення з датасету та згенерованого LoRA-моделю №3

На лівому зображенні (рис. 3.12) з датасету для навчання моделі LoRA я знятий знизу. Мій погляд виглядає наляканим або трохи стривоженим. Я в темній футболці, а фон сірий і текстурований.

На правому зображенні (рис. 3.12), згенерованому за допомогою моделі LoRA, мене зображено з такого ж ракурсу, як і на зображенні з датасету. Однак тут я виглядаю спокійнішим. Крім того, згенероване зображення має краще освітлення. Фон схожий на оригінальний, але відрізняється за кольором.

Зображення досить схожі, але як і у попередніх випадках LoRA-модель спотворила емоцію на обличі. Також на зображенні, що згенерувала модель, обличчя набагато краще освітлено.

У цьому підрозділі було проведено порівняння між зображеннями з датасету та згенерованими за допомогою LoRA-моделі на основі текстових описів. Аналіз трьох пар зображень дозволив зробити наступні спостереження:

а) Відтворення зовнішнього вигляду та емоцій: Зображення з датасету мають більш природний вигляд, тоді як згенеровані зображення демонструють відмінності у виразі обличчя та настрої. LoRA-модель часто змінює емоції, створюючи більш виражені або напружені вирази.

б) Фон та освітлення: Оригінальні зображення мають нейтральні або текстуровані фони, тоді як згенеровані зображення часто мають більш кінематографічний характер завдяки деталізованому освітленню та зміненому фону. Це робить згенеровані зображення більш виразними, але менш природними.

в) Деталізація та ракурси: Хоча згенеровані зображення відображають загальні риси оригіналів, вони часто виглядають більш художньо обробленими, з покращеним освітленням та деталізацією. Це надає їм естетичної привабливості, але водночас віддаляє від природного вигляду.

Отже, порівняння показало, що зображення, згенеровані за допомогою LoRA-моделі, хоча й мають свої переваги у вигляді покращеного освітлення та деталізації, не завжди точно відтворюють емоційний стан та природний вигляд оригінальних зображень. Це свідчить про певні обмеження моделі в точності

відтворення вихідних характеристик, але також вказує на її потенціал у створенні виразних та естетично привабливих зображень.

ВИСНОВКИ

Незважаючи на наявний досвід роботи з моделями LoRA, їхні принципи функціонування та методи навчання залишалися «незвіданою територією». Вивчення розпочалося з методів генерації зображень, таких як GAN, VAE та Stable Diffusion, зосереджуючись на аналізі їхньої структури та ключових особливостей. При цьому складні математичні формули були оминути, а пояснення надавалися на зрозумілих прикладах.

Було визначено, що для роботи з LoRA-моделями найкраще підходять саме моделі стабільної дифузії (Stable Diffusion). Після цього розглянуто, що таке LoRA і як вона використовується, на конкретних прикладах. Щоб зрозуміти структуру і принципи роботи LoRA, довелося зануритися в математичні аспекти, детально пояснюючи кожен принцип, хоча ця тема залишається досить складною.

Переваги та недоліки LoRA було оцінено, і перед тим, як перейти до навчання моделі, досліджено веб-інтерфейси програм для генерації зображень і навчання моделей. Порівнявши старі та нові моделі Stable Diffusion, з'ясувалося, що нові версії значно краще виконують завдання генерації зображень, тому їх і використано для навчання LoRA-моделі.

Процес підготовки датасету з зображень і текстових описів, структуру папок і налаштування у веб-інтерфейсі було детально описано. Після всіх підготовчих робіт було запущено процес навчання LoRA-моделі. Після завершення навчання отримано модель, яка генерувала зображення обличчя. Досліджено процес генерації зображень з шуму, використовуючи Sampling steps та Seed, і проаналізовано вплив семплерів на результати.

Згенерувавши зображення за допомогою власної LoRA-моделі та іншої моделі, знайденої у відкритому доступі, результати порівняно за різними промтами. Також спробувано згенерувати зображення, використовуючи текстовий опис зображень з датасету для навчання LoRA-моделі, і порівняно їх з оригіналами.

У підсумку вдалося створити LoRA-модель, яка генерувала зображення обличчя. Хоча сподівалася на кращу якість зображень, для моделі, навченої лише на 16 зображеннях, це досить непоганий результат. Отримано багато нових знань про ШІ, зокрема, дізнанося, що великі мовні моделі, такі як ChatGPT, можуть бути донавчені за допомогою LoRA-моделей.

Ця робота дозволила поглибити знання у сфері штучного інтелекту, від принципів генерації зображень і підготовки даних до тонких налаштувань моделей. Окремо варто відзначити компанію Huawei Україна, яка надала лекційну базу для переддипломної практики. Ці знання можуть допомогти стати кваліфікованим фахівцем у цій галузі.

Під час роботи з LoRA-моделями стало зрозуміло, що вони мають величезний потенціал. Вони здатні не лише копіювати об’єкти з датасету, але й відтворювати різні художні стилі, що робить їх потужним інструментом для художників, дизайнерів, фотографів, артистів і маркетологів. Використовуючи LoRA-моделі, можна експериментувати, творити та надихатися.

Модель була навчена лише на 16 зображеннях, але є переконання, що використання більшого та якіснішого датасету суттєво покращить результати.

Завершуючи проєкт, варто підкреслити, що індустрія штучного інтелекту швидко розвивається і змінюється, а LoRA-моделі займають у ній важливу нішу.

ПЕРЕЛІК ПОСИЛАНЬ

1. 3Blue1Brown. But what is a neural network? | Chapter 1, Deep learning, 2017. YouTube. URL: https://www.youtube.com/watch?v=aircAruvnKk (date of access: 22.05.2024).
2. 3Blue1Brown. Gradient descent, how neural networks learn | Chapter 2, Deep learning, 2017. YouTube. URL: https://www.youtube.com/watch?v=IHZwWFHWa-w (date of access: 22.05.2024).
3. 3Blue1Brown. What is backpropagation really doing? | Chapter 3, Deep learning, 2017. YouTube. URL: https://www.youtube.com/watch?v=Ilg3gGewQ5U (date of access: 22.05.2024).
4. AI Coffee Break with Letitia. What is LoRA? Low-Rank Adaptation for finetuning LLMs EXPLAINED, 2023. YouTube. URL: https://www.youtube.com/watch?v=KEv-F5UkhxU (date of access: 22.05.2024).
5. Automatic1111. Open Source Initiative. Version 1.9.3. GitHub, 2023. URL: https://github.com/AUTOMATIC1111/stable-diffusion-webui
6. HCIA-AI V3.5 course. Huawei Enterprise | Accelerate Industrial Intelligence. URL: https://e.huawei.com/en/talent/outPage/#/sxz-course/home?courseId=LEr4w0UtMYfpbhkgY_Yd5QqFAO0 (date of access: 22.05.2024).
7. Hu E. Lora: low-rank adaptation of large language models. 2021. 26 p. URL: https://arxiv.org/pdf/2106.09685.
8. Introducing LoRA: A faster way to fine-tune Stable Diffusion. Replicate – Run AI with an API. URL: https://replicate.com/blog/lora-faster-fine-tuning-of-stable-diffusion (date of access: 22.05.2024).
9. Kohya_ss. Open Source Initiative. Version 24.1.4. GitHub, 2023. URL: https://github.com/bmaltais/kohya_ss.

10. Li C. Measuring the intrinsic dimension of objective landscapes. *arXiv.org*. URL: https://arxiv.org/abs/1804.08838 (date of access: 22.05.2024).
11. LoRA. Hugging Face – The AI community building the future. URL: https://huggingface.co/docs/diffusers/training/lora (date of access: 22.05.2024).
12. LoRA, low-rank adaptation, large foundation models, fine-tuning, AI trends, computational resources. ML6: Accelerate Intelligence | AI Solutions & Strategy. URL: https://www.ml6.eu/blogpost/harnessing-the-power-of-foundation-models-for-your-business-with-lora (date of access: 22.05.2024).
13. Lora stable diffusion AI models | civitai. *Civitai: The Home of Open-Source Generative AI*. URL: https://civitai.com/tag/lora (date of access: 22.05.2024).
14. Nagaraj N. Low rank adaptation: a technical deep dive. *Medium*. URL: https://blog.ml6.eu/low-rank-adaptation-a-technical-deep-dive-782dec995772 (date of access: 22.05.2024).
15. Sg_161222. Realistic vision V6.0 B1 - V5.1 hyper (VAE) | stable diffusion checkpoint | civitai. Civitai: The Home of Open-Source Generative AI. URL: https://civitai.com/models/4201?modelVersionId=501240 (date of access: 22.05.2024).
16. Variational lossy autoencoder. https://openai.com/. URL: https://openai.com/index/variational-lossy-autoencoder/ (date of access: 22.05.2024).
17. What is a GAN? - generative adversarial networks explained - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is/gan/?nc1=h_ls (date of access: 22.05.2024).
18. 18. What is generative AI? - generative artificial intelligence explained - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is/generative-ai/?nc1=h_ls (date of access: 22.05.2024).
19. What is stable diffusion? - stable diffusion AI explained - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is/stable-diffusion/?nc1=h_ls (date of access: 22.05.2024).

20. Zhao G. How Stable Diffusion works, explained for non-technical people, 2023. Medium. URL: https://bootcamp.uxdesign.cc/how-stable-diffusion-works-explained-for-non-technical-people-be6aa674fa1d (date of access: 22.05.2024).

ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ (Презентація)

1

Державний університет інформаційно-комунікаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

«Створення LoRA моделі для генерації зображень за допомогою текстових запитів»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Виконав(ла): Панченко В.Ю, ІСД-42
Науковий керівник роботи:
Каграманова Ю. К.

Київ - 2024

2

Актуальність теми. Зростаючий попит на генерацію зображень зумовлений розвитком технологій та змінами у споживчих звичках. Соціальні мережі, електронна комерція та мультимедійні платформи вимагають унікального візуального контенту. У цьому контексті LoRA-модель для генерації зображень за текстовими запитами має великий потенціал.

Об'єкт дослідження – процес створення LoRA-моделі.

Предмет дослідження. Порівняння методів генерації, архітектуру LoRA-моделей, підготовку даних, навчання, впровадження та оцінку якості зображень.

Мета та завдання дослідження. Метою цього дослідження є створення моделі LoRA, яка базується на великій моделі генерації зображень та наборі даних, що складається з фотографій та текстових файлів з їхніми описами, з метою навчання моделі.
З метою реалізації мети дослідження було сформульовано наступні завдання:
а) Порівняти та проаналізувати існуючі методи генерації зображень, зокрема GAN, VAE та моделі дифузії, з метою виявлення їхніх переваг та недоліків.
б) Описати архітектуру та принципи роботи LoRA-моделей для генерації зображень, визначити їх ключові переваги та оцінити обмеження.
в) Підготувати дані для навчання моделі, включаючи збір та підготовку текстових та візуальних даних.
г) Провести процес навчання LoRA-моделі для генерації зображень.
д) Провести якісну оцінку згенерованих зображень.
е) Узагальнити результати дослідження та виокремити ключові висновки щодо ефективності LoRA-моделі, пояснити практичну користь та можливості використання розробленої моделі, а також визначити перспективні напрямки для подальшого розвитку дослідження.

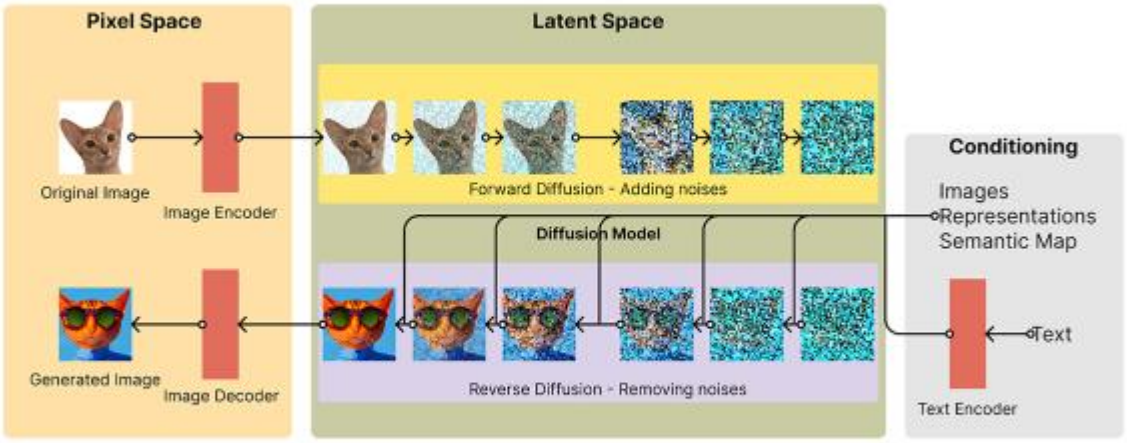


Схема роботи методу генерації зображень Stable Diffusion

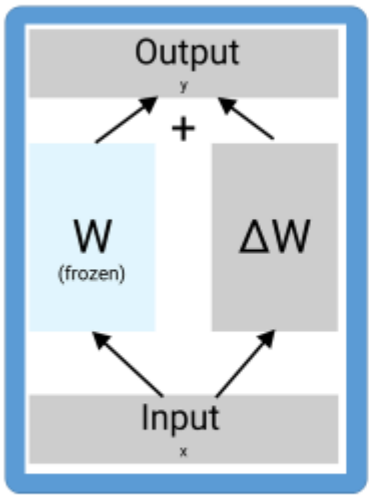
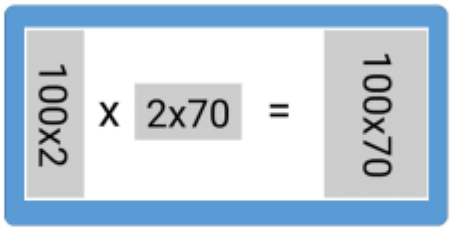


Схема процесу навчання моделі ΔW



Приклад для розмірність вхідного x вихідного простору 100 x 70

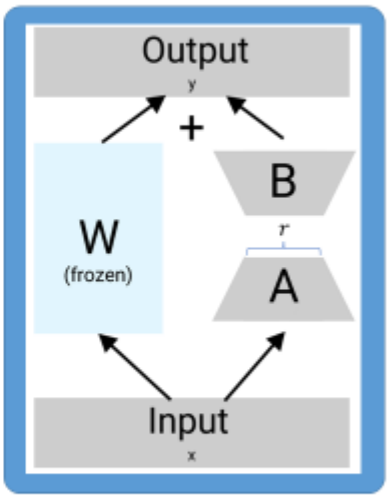
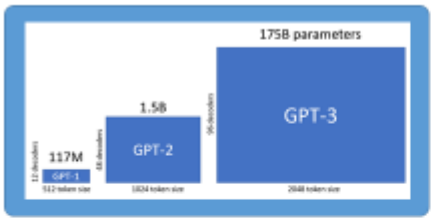
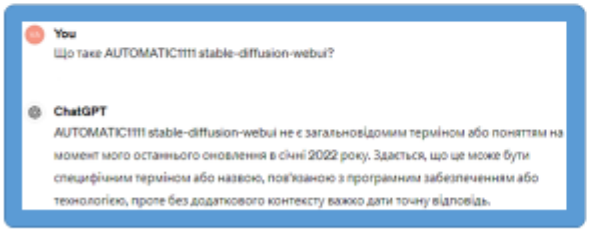


Схема процесу розбиття матриці на менші матриці A і B

LoRA-моделі: огляд та принципи роботи



Кількість параметрів великих мовних моделей



Відповідь GPT-3 на запит, що стосується події, яка сталася після 2022 року

5

Переваги та недоліки LoRA-моделей

Переваги:

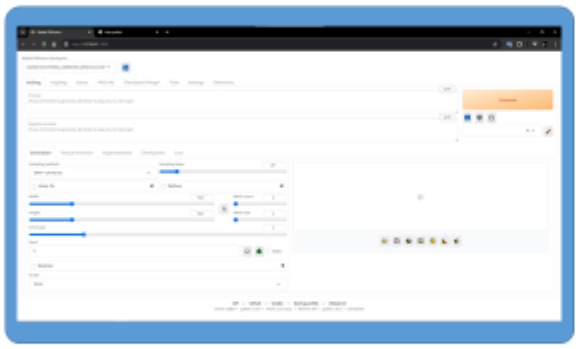
- Значно дешевше донавчання
- Економія місця на диску
- Відсутність затримок при висновках
- Динамічна зміна стилю

Недоліки:

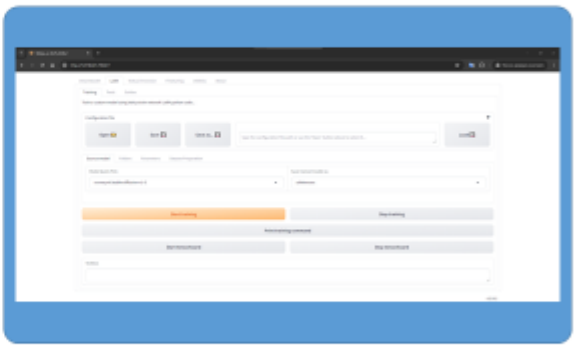
- Складність реалізації
- Залежність від базової моделі
- Потенційна втрата точності
- Обмежена гнучкість
- Недостатня прозорість

6

РОЗРОБКА LoRA МОДЕЛІ

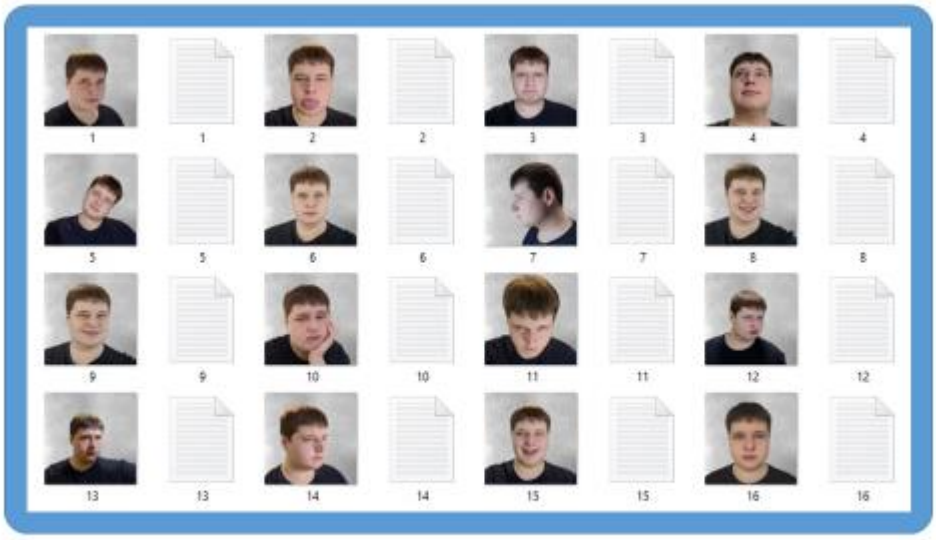


Вигляд веб-інтерфейсу A1111



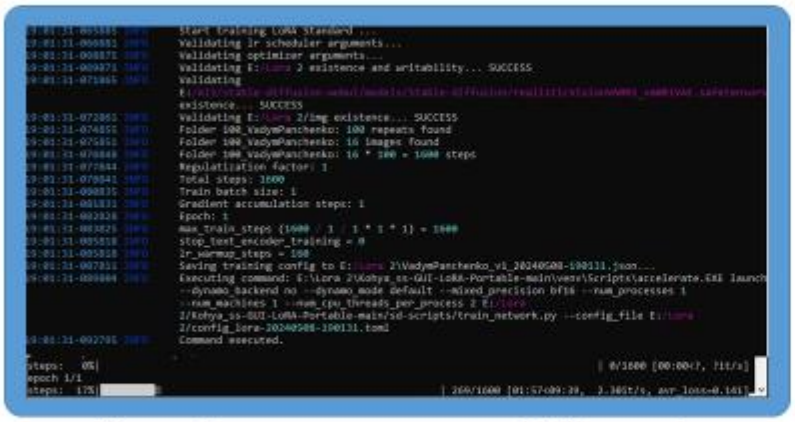
Вигляд веб-інтерфейсу Kohya_ss

7



Вигляд датасету з зображеннями та текстовим описом до них

8



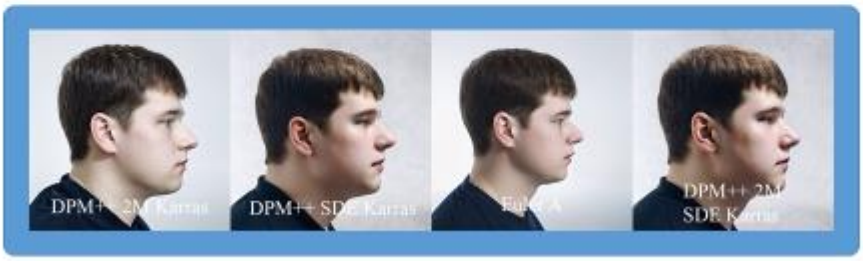
Консоль khoya_ss у момент навчання LoRA-моделі

Створена LoRA-модель

9

ТЕСТУВАННЯ ТА ОЦІНКА МОДЕЛІ

Вплив Sampling steps на якість згенерованого зображення



Вплив семплерів на зображення, що генерується

10

Використання моєї LoRA-моделі

Поріняння зображення з датасету та згенерованого LoRA-моделю

11

ВИСНОВКИ

Вивчення розпочалося з методів генерації зображень, таких як GAN, VAE та Stable Diffusion. Далі було визначено, що таке LoRA, її призначення та принципи роботи. З'ясовано, що для роботи з LoRA найкраще підходять моделі стабільної дифузії. Проведено аналіз середовищ для генерації зображень та навчання LoRA-моделей. Після підготовки датасету з 16 зображень та текстових описів розпочато процес навчання LoRA-моделі.

Результатом роботи стала LoRA-модель, яка успішно генерувала зображення обличчя. Хоча якість зображень можна покращити, результат є задовільним, враховуючи обмежений обсяг даних для навчання. Порівняння зображень, згенерованих цією моделлю та іншими моделями, показало великий потенціал для створення зображень за текстовими описами.

Цей проєкт дозволив мені значно поглибити знання в області штучного інтелекту і LoRA-моделей. Я зрозумів, що LoRA-моделі мають величезний потенціал у різних галузях, таких як мистецтво, дизайн та маркетинг. Дякую компанії Huawei Україна за лекційну базу, яка допомогла розширити мої знання і навички.

12

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

V Міжнародна науково-технічна конференція «Сучасний стан та перспективи IoT», яка проходила 18 квітня 2024 року. Тези на тему «Вплив відеокарт на ШІ» та «Оцінка якості моделі ШІ» було опубліковано у збірнику, присвяченому цій конференції.

Дякую за увагу!