**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ при Президенте Российской Федерации»**

**ИНСТИТУТ ЭКОНОМИКИ, МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**ЭКОНОМИЧЕСКИЙ ФАКУЛЬТЕТ**

**НАПРАВЛЕНИЕ 38.03.01 ЭКОНОМИКА**

|  |
| --- |
|  |

**Проект в рамках курса «Глубокое обучение»**

**Генерация изображений в стиле Studio Ghibli.**

**Проект по созданию кастомной диффузионной модели с LoRA.**

|  |  |
| --- | --- |
| студент-бакалавр  Посохова Ксения Михайловна  Сергеева Екатерина Дмитриевна |  |
| Преподаватель  Чухров Никита Сергеевич |  |

**МОСКВА**

**2025**

**Введение**

В рамках проектной деятельности на курсе глубокого обучения мы выбрали тему **«**Условная генерация изображений**»**.

Нашей целью стало обучение модели *Stable Diffusion 2.1* создавать изображения в уникальном и узнаваемом стиле *студии Ghibli*. Этот стиль популярен и является уже классическим в мире анимации. Он обладает ярко выраженными стилистическими особенностями, которые может обнаружить модель и создавать новые изображения с узнаваемыми чертами лиц персонажей, проработкой фона и цветовых аспектов.

Для дообучения модели мы использовали технику *LoRA (Low-Rank Adaptation)* — подход, который позволяет адаптировать большие нейросети, изменяя лишь небольшую часть параметров. Такой подход помогает оптимизировать использование ограниченных ресурсов и ускоряет обучение, сохраняя при этом качество результата.

Мы собрали датасет из изображений с официального сайта студии Гибли, сопроводили их описаниями с помощью Blip модели, внедрили в словарь специальный токен <ghibli> и провели дообучение модели.

Финальную версию развернули через **Gradio**, чтобы любой пользователь мог ввести текст и получить изображение в стиле Ghibli.

Этот проект позволил нам поработать с диффузионными моделями, понять, как работает дообучение в случае генерации изображений и сделать кучу смешных/милых/красивых картинок.

**Основная** **часть**

1. Цель проекта

Наш проект направлен на решение задачи **условной генерации изображений по текстовому описанию**. Мы взяли за основу мощную модель Stable Diffusion 2.1 — это генеративная диффузионная модель, способная создавать высококачественные изображения из текста. Мы решили дообучить её для генерации картинок в стиле **студии Ghibli**, известной по фильмам Хаяо Миядзаки.

Главная идея проекта — добиться того, чтобы по текстовому описанию, содержащему токен <ghibli> или прямое упомненные стиля Ghibli, модель генерировала изображения, визуально соответствующие эстетике японской анимационной студии.

2. Сбор и обработка данных

Мы решили создать свой собственный датасет, состоящий из кадров фильмов студии Ghibli. Всего мы собрали 250 изображений из 5 анимационных проектов:

* Принцесса Мононоке
* Мой сосед Тоторо
* Унесённые призраками
* Ходячий замок
* Мальчик и птица

Все кадры были взяты с официального сайта Ghibli.

Далее перед нами встала задача написания текстового описание к собранным изображениям. Поскольку вручную писать было бы слишком долго, мы приняли решение использовать готовую нейронную сеть из пакета transformers Blip base. Она с работой справилась, однако в ней было слишком много недочётов, поэтому мы решили использовать её расширенную версию Blip large. Также мы рассматривали более современную модель Blip2, но, к сожалению, наши ресурсы не позволяли с ней работать. Таким образом наш выбор остановился на Blip large.

После использования выбранной image-to-text модели, нам было необходимо просмотреть полученный результат и при необходимости его подкорректировать. Для этого мы написали скрипт, который выдавал нам поочерёдно все 250 изображений и в случае неправильного написания позволял исправить ошибку. Написание этого скрипта ускорило процесс проверки.

Выбранная нами модель для дообучения принимает на вход изображения определённого формата, а именно 512×512 пикселей. Наши же изображения были 1038×1920 пикселей. Мы решили не обрезать изначальные изображения, а уменьшить их и при помощи падинга привести их в квадратную форму. Так мы пытались не потерять важную информацию с изображений.

Тут мы также добавляем дополнения для текста, а именно токен <ghibli>, который позволит нашей модели отличать запросы на такой стиль.

3. Обучение модели

На этапе обучения мы протестировали разные методы, в том числе и следующий подход: генерация изображений на каждой эпохе обучения по единому промту (наш был следующий: «a boy with a bouquet of flowers in <ghibli> style». Благодаря этому мы смогли найти точку, когда изображения начинают сильно проседать в качестве – этап переобучения, модель начинает «забывать» изначальную информацию. Так от 10 эпох мы пришли к 4 эпохам. При этом использовали батч = 1, т.к. мощности не позволяли поставить большее значение.

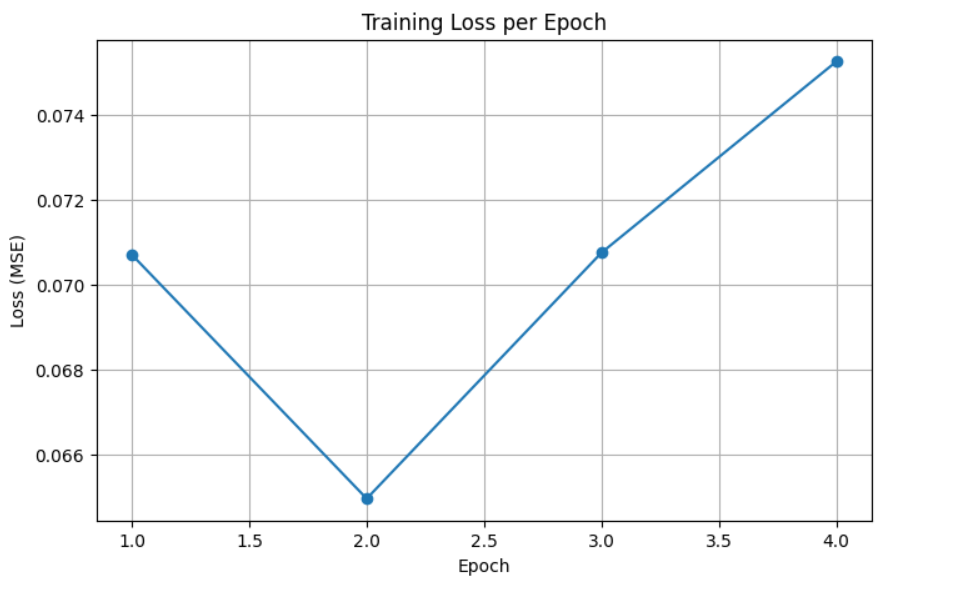
Также мы тестировали разные методы предобработки данных, в том числе заметили, что если просто обрезать изображения, то из-за потери информации, мы теряем качество, поэтому решили оставить паддинг.

Основные компоненты:

* VAE (AutoencoderKL) — преобразует изображение в сжатое латентное пространство и обратно.
* Text Encoder (CLIP) — преобразует текстовое описание в векторное представление.
* UNet с LoRA — учится предсказывать добавленный шум, и именно его веса дообучаются.
* Noise Scheduler (DDPMScheduler) — задает последовательность шагов и степень шума, добавляемого на каждом этапе.

Процесс обучения по шагам:

1. **Подготовка батча данных**  
   Из каждого батча извлекаются изображения (pixel\_values) и соответствующие текстовые описания, уже содержащие токен <ghibli>.
2. **Преобразование изображений в латенты**  
   Изображения кодируются с помощью VAE в компактное латентное пространство. Это делается без градиентов, так как VAE остаётся замороженным (не обучается).
3. **Добавление случайного шума**  
   К полученным латентам добавляется случайный шум, соответствующий случайному времени (timestep) — это имитирует процесс деградации изображения в диффузионной модели.
4. **Получение текстовых эмбеддингов**  
   Текстовое описание пропускается через CLIP Text Encoder. Его выходные эмбеддинги используются в UNet как контекстная информация.
5. **Предсказание шума с помощью UNet**  
   UNet (с интегрированными адаптерами LoRA) пытается предсказать шум, добавленный к латенту. Для этого используется контекст в виде текстовых эмбеддингов и текущий шаг по времени.
6. **Вычисление ошибки (MSE Loss)**  
   Потери рассчитываются как среднеквадратичное отклонение между предсказанным и настоящим шумом.
7. **Обновление весов LoRA**  
   Используется AdamW-оптимизатор и Cosine Learning Rate Scheduler. Для повышения производительности включен автоматический mixed precision (AMP) через GradScaler.
8. **Финальная генерация и сохранение**  
   После завершения всех эпох дообученная LoRA-секция сохраняется отдельно. Также выполняется финальная генерация изображения по заданному тексту, чтобы продемонстрировать итоговый результат.
9. **Визуализация обучения**  
   Строится график значений потерь (Loss) по эпохам, чтобы визуально оценить динамику сходимости модели. График лосса достаточно нестабилен, т.к. всего 1 батч, мало данных для дообучения модели и т.д. Однако в случае нашей задачи это нормально.



4. Создание оболочки (интерфейса)

После завершения обучения модели и сохранения весов LoRA, следующим этапом стало создание пользовательского интерфейса — оболочки, с помощью которой можно удобно взаимодействовать с моделью и генерировать изображения.

Для этой задачи мы использовали библиотеку **Gradio**, так как она предоставляет простой способ развернуть веб-интерфейс для моделей машинного обучения без необходимости ручного создания сайта или настройки серверной логики. Gradio позволяет буквально за несколько строк кода организовать полноценное приложение, к которому можно подключиться через браузер.

Этапы создания оболочки:

1. **Загрузка обученной модели**  
   Сначала мы загружаем Stable Diffusion Pipeline с подключённой LoRA-моделью.
2. **Определение функции генерации**  
   Мы определили функцию, которая принимает текстовый промпт от пользователя, передаёт его в пайплайн и возвращает сгенерированное изображение.
3. **Настройка Gradio-интерфейса**  
   Мы создали простую форму с одним текстовым полем и кнопкой генерации. После ввода текста, пользователь нажимает кнопку, и результат отображается прямо в браузере.

**Заключение**

В ходе выполнения проекта мы успешно реализовали задачу условной генерации изображений в стиле студии Ghibли на базе модели Stable Diffusion 2.1 с использованием техники дообучения LoRA. Мы собрали и подготовили качественный датасет, адаптировали модель к специфике выбранного художественного стиля и провели экспериментальное обучение, что позволило добиться генерации узнаваемых и эстетически привлекательных изображений.

Использование LoRA позволило эффективно дообучать модель при ограниченных вычислительных ресурсах, сохранив при этом высокое качество результатов. Созданный интерфейс на базе Gradio делает наш проект доступным и удобным для пользователей, что расширяет возможности применения данной технологии для творческих и практических целей.

Реализация проекта дала нам глубокое понимание принципов работы диффузионных моделей и методов адаптации больших нейросетей под конкретные задачи, а также позволила освоить инструменты для создания интерактивных приложений машинного обучения.

В перспективе можно продолжить работу над улучшением качества генерации, увеличением разнообразия стиля и расширением функционала интерфейса, что сделает проект ещё более полезным и интересным.

P.S. а еще захотелось пересмотреть все мультипликации студии Гибли 😊