Генерация логотипов бренда по текстовому описанию

Команда проекта: Богдан, Александр, Антон, Роман
Куратор: Ева Неудачина

2 января 2025 г.

Содержание

1 Описание проекта

Данный проект направлен на создание системы, генерирующей логотипы на основе текстовых описаний. Пользователь вводит текстовое описание, и система создает изображение, соответствующее этому описанию. На данном этапе в проекте используется генеративная состязательная нейросеть (GAN), которая обучена на большом датасете с изображениями логотипов и их текстовыми описаниями.

2 Структура проекта

Проект состоит из нескольких частей:

- Обучение модели GAN на базе датасета логотипов.
- Создание API для генерации изображений.
- Разработка Streamlit-приложения для удобного взаимодействия с моделью.

3 Функционал API

Эта часть проекта представляет собой API (Application Programming Interface), который выступает в роли прослойки между Streamlit-интерфейсом и сервером. Основная задача API — принимать запросы от пользователя, обрабатывать данные на сервере и возвращать результат обратно в приложение. В данном случае API реализует одну ключевую функцию: генерацию логотипа по текстовому описанию

1. Принимает запросы от приложения
    - Пользователь через Streamlit-приложение отправляет POST-запрос с текстовым описанием логотипа
    - API принимает этот запрос и извлекает данные для дальнейшей обработки
2. Обрабатывает данные на сервере
    - Текстовое описание обрабатывается
    - Данные передаются в генеративную модель, которая генерирует логотип
3. Возвращает результат приложению
    - API возвращает сгенерированное изображение в формате JSON, которое может быть отображено в Streamlit-приложении
    - Streamlit-приложение получает изображение и отображает его пользователю

4 Функционал Streamlit-приложения

Streamlit-приложение предоставляет пользователю интерфейс для работы с моделью и анализа данных для датасета, аналогичного тому, который использовался для обучения GAN. Оно включает следующие секции:

1. Генерация изображений: пользователь вводит текстовое описание, после чего генерируется 4 варианта логотипов.
2. Краткая информация о модели: отображаются метрики качества, графики потерь и информация о модели.
3. Загрузка датасета: пользователь может загрузить датасет, состоящий из изображений и их текстовых описаний, для анализа.
4. EDA (Exploratory Data Analysis): позволяет анализировать текстовые и графические данные, сгенерированные из загруженного датасета, с возможностью интерактивного выбора графиков и настроек.

Пример интерфейса Streamlit-приложения до загрузки данных и использования представлен на Рис. 1. Чтобы ознакомиться с записью примеров использования приложения, пройдите по ссылке в репозиторий проекта.

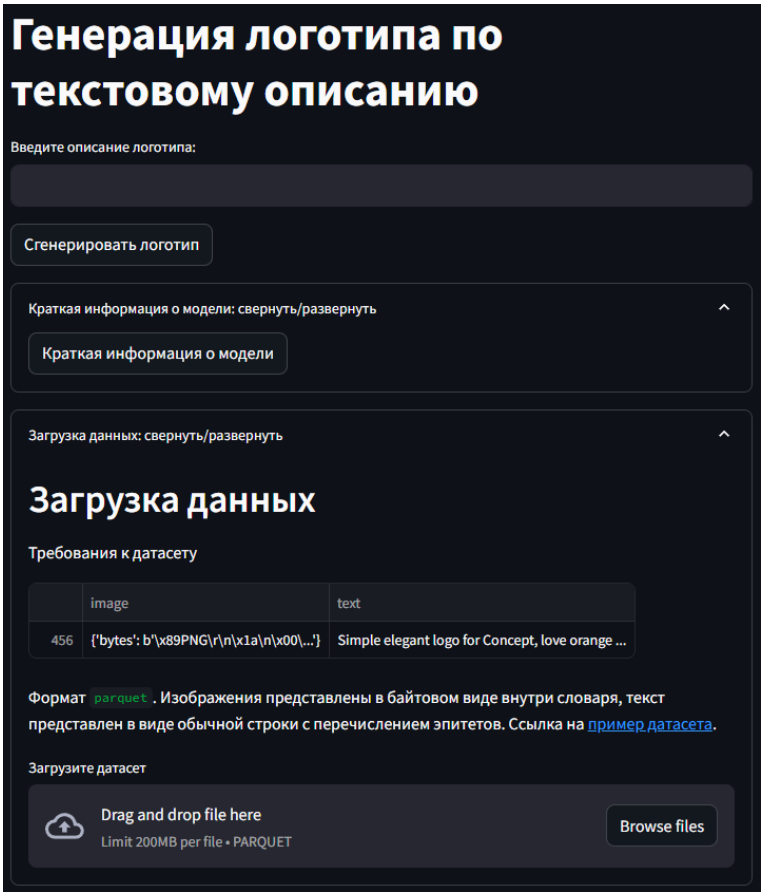

Рис. 1: Интерфейса Streamlit-приложения.

5 Развертывание сервера на платформе Selectel.ru

Проект был успешно развернут на созданном сервере компании Selectel, предоставляющей облачные инфраструктурные сервисы и услуги дата-центров. Ниже представлены основные этапы развертывания сервера.

5.1 Этапы развертывания сервера

5.1.1 Разработка онлайн-сервера в конструкторе, подбор оптимальных параметров сервера

Используя конструктор серверов от Selectel, был разработан сервер, соответствующий требованиям проекта. Проведен анализ и выбор оптимальных параметров сервера с учетом соотношения цена/качество. Обратите внимание, что реализация сервера является платной.

5.1.2 Запуск сервера

После завершения настройки параметров, сервер был запущен и готов к дальнейшим действиям.

5.1.3 Установка необходимых инструментов

- Установка Git:

```
sudo apt update
sudo apt install git
```

Код 1: Установка Git.

- Установка Docker:

```
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
↪  https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Код 2: Установка Docker.

Данные команды осуществляют:

- Установку сертификатов и curl;
- Создание каталога для хранения ключей;
- Загрузку GPG-ключа Docker (используется для проверки подлинности пакетов из репозитория Docker);
- Настройку прав доступа на ключ;
- Добавление репозитория Docker в источники APT;
- Обновление списка пакетов.

5.1.4 Клонирование проекта из Git на сервер

```
git clone https://github.com/HerrVonBeloff/AI-YP_24
```

Код 3: Клонирование проекта из Git на сервер.

5.1.5 Запуск Docker-билда

```
cd /HerrVonBeloff/AI-YP_24
docker-compose up -d --build
```

Код 4: Запуск Docker-билда.

6 Инструкция по использованию

6.1 Локальный запуск с помощью Docker

Для локального использования и тестирования приложения необходимо склонировать репозиторий, затем локально в папке с репозиторием выполнить следующую команду:

```
sudo docker-compose up -d --build
```

Код 5: Первый пример команды Docker Compose.

После того, как данная команда будет выполнена, необходимо выполнить:

```
sudo docker-compose logs
```

Код 6: Второй пример команды Docker Compose.

В ходе выполнения команды будет выведена информация с ссылками на Streamlit-приложение (Код. 7).

```
streamlit   |  You can now view your Streamlit app in your browser.
streamlit   |
streamlit   |  Local URL: http://localhost:8501
streamlit   |  Network URL: http://172.19.0.3:8501
streamlit   |  External URL: http://95.24.77.148:8501
```

Код 7: Адрес Streamlit-приложения.

Документацию FastApi можно будет посмотреть по адресу: http://localhost:8000/docs.

6.2 Запуск с VPS-сервера

Приложение можно использовать по адресу: http://46.161.52.173:8501/.

6.3 Streamlit-приложение

Streamlit-приложение предоставляет пользователю интерфейс для работы с моделью и анализа данных для датасета, аналогичного тому, который использовался для обучения GAN. Оно включает следующие секции:

1. Генерация изображений. Пользователь вводит описание, и модель генерирует логотип. Осуществляется после ввода текстового описания изображения в специальное поле и нажатия кнопки `Сгенерировать логотип`. Взаимодействвие с моделью осуществляется через API.

2. Краткая информация о модели. Даёт краткую справку о модели, выводит интерактивные графики для значений потерь и метрики FID, даётся ссылка на репозиторий для ознакомления с боле подробной информацией и кодом проекта. Пользователь может выбрать цветовую тему для интерактивных графиков из выпадающего меню в данном разделе.

3. Загрузка датасета. Позволяет загружать датасет формата `parquet`, аналогичный тому, что использовался при обучении моделей, в Streamlit-приложение для анализа данных. Датасет должен состоять минимум из двух колонок с названиями `image` и `text`, в которых ханятся изображения и их текстовое описание соответственно. При этом изображение в каждом объекте должно быть представлено в виде словаря с ключом `bytes`, в значении которого содержится изображение в байтовом представлении (Таблица 1).

	image	text
456	{'bytes': <байтовая строка>}	Simple elegant logo for Concept, ...

Таблица 1: Пример объекта датасета с индексом 456.

После загрузки датасета появляется кнопка `Получить случайный элемент датасета`, при нажатии на которую выводится случайный логотип из датасета и его текстовое описание с указанием индекса.

Пример датасета, который представляет собой срез обучающей выборки для нашей модели, представленный двумя тысячами его последних элементов, можно скачать по ссылке. Ссылка на пример датасета и описание требований к нему даны и в самом web-приложении.

4. EDA. Выводит анализ данных загруженного датасета. Кнопка `EDA` появляется только после загрузки датасета в Streamlit-приложение. Выводит следующие графики:

   - Анализ текстовых данных
     - Облако слов
     - Гистограмма распределения длин описаний (в символах)
     - Boxplot для длин описаний
     - Гистограмма количества эпитетов в описании
     - Boxplot для количества эпитетов
   - Анализ изображений
     - Соотношение RGB и чёрно-белых логотипов (круговая диаграмма)
     - Высота изображений (гистограмма)
     - Ширина изображений (гистограмма)
     - Соотношение сторон (гистограмма)
     - Соотношение сторон (без квадратных изображений, гистограмма)
     - Количество пикселей (гистограмма)

   Данные графики дают начальное представление о данных и позволяют наметить первые шаги к их предобработке. Каждый график, кроме облака слов, является интерактивным. Для каждой гистограммы можно настроить количество бинов для удобства восприятия пользователя. Помимо этого, предоставляется выпадающее меню, позволяющее выбрать цветовую тему для интерактивных графиков из списка тем библиотеки `plotly`.

6.4 Использование API

Для использования API отправьте запрос с текстовым описанием в Streamlit, и получите сгенерированное изображение.

7 Заключение

Этот проект предоставляет пользователю инструмент для генерации логотипов на основе текстового описания. Система использует современную технологию генеративных нейросетей и предоставляет удобные интерфейсы для пользователей. На данный момент, так как это baseline, модель еще требует доработки и вычислительных ресурсов для дальнейшего улучшения.