

Text-to-Image React Web App

NLP Web Final Report

F07942100, Wu,Yuan-Kuei, 電信所博三
F07942089, 劉廷緯, 電信所博三

Motivation

“Text-to-Image” machine learning models are not easily accessible to the everyday user. The theory behind the model is hard to understand, and running the code itself is challenging. Users will have to handle machine learning libraries like PyTorch, install drivers like CUDA, and prepare GPU hardware for the model’s inference. With this in mind, we solve the problem by providing a web app as a user interface to the “Text-to-Image” model. We designed our web app to adopt the “stable diffusion” model, allowing everyone easy access to the state-of-the-art text-to-image “stable diffusion” model. The “stable diffusion” model is a latent text-to-image diffusion model capable of generating photo-realistic images given any text input. For example, given the following input, “a photograph of an astronaut riding a horse”, our website will generate the following:



Impact

Our “Text-to-Image React Web App” allows users to generate photo-realistic images given any text input. Also, the users will have parameters to choose from, for example, positive text input (prompt), negative text input (negative prompt), width, height, number of inference steps, number of images to output, etc. The most significant contribution of our web app is that we wrapped the machine learning model into a web interface. With our web app, users no longer have to deal with challenging technical details but can directly use the machine learning model through our web interface. We also provide GPU inference on our remote server, so the

computation is done at our end and not on the user's side. We show the interface of our web app in the following image:

Stable Diffusion

Prompt *

There is a pig.

Negative Prompt *

Width

512

Width of output image. Maximum size is 1024x768 or 768x1024 because of memory limits

Height

512

Height of output image. Maximum size is 1024x768 or 768x1024 because of memory limits

num_inference_steps

50


Number of denoising steps (minimum: 1; maximum: 500)

Num

1

Number of images to output. (minimum: 1; maximum: 4)

SUBMIT



Design

Website front-end: React

Website backend: Flask (A lightweight web application framework written in Python)

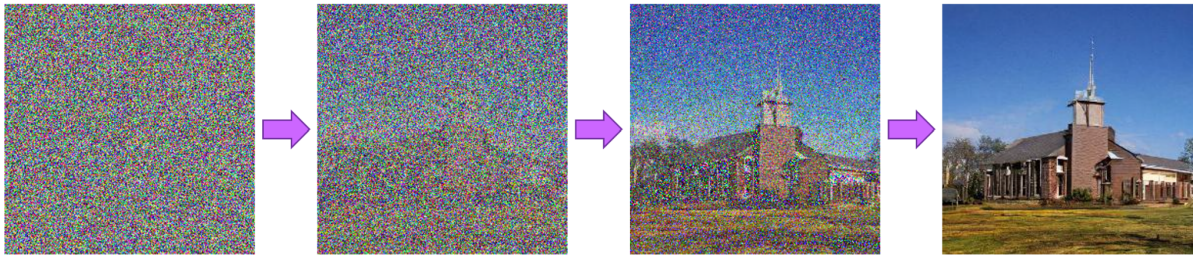
Workflow:

- Enter prompt and other parameters (React -> Flask -> Model)
- Run text-to-image and save it to storage (Model -> Storage; Flask -> React)
- React retrieves the generated image from remote (Storage -> React)

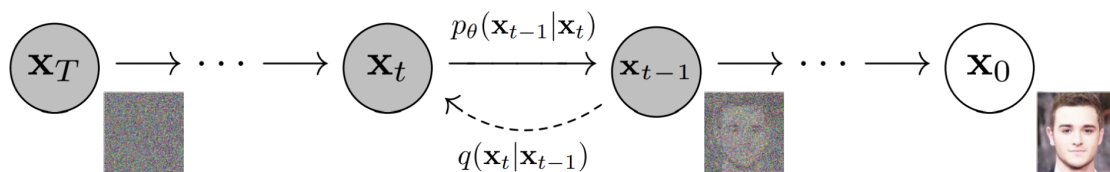
The “stable diffusion” requires a powerful GPU to run inference, so we run it on our own remote server and forward the webpage to the demo PC.

Method

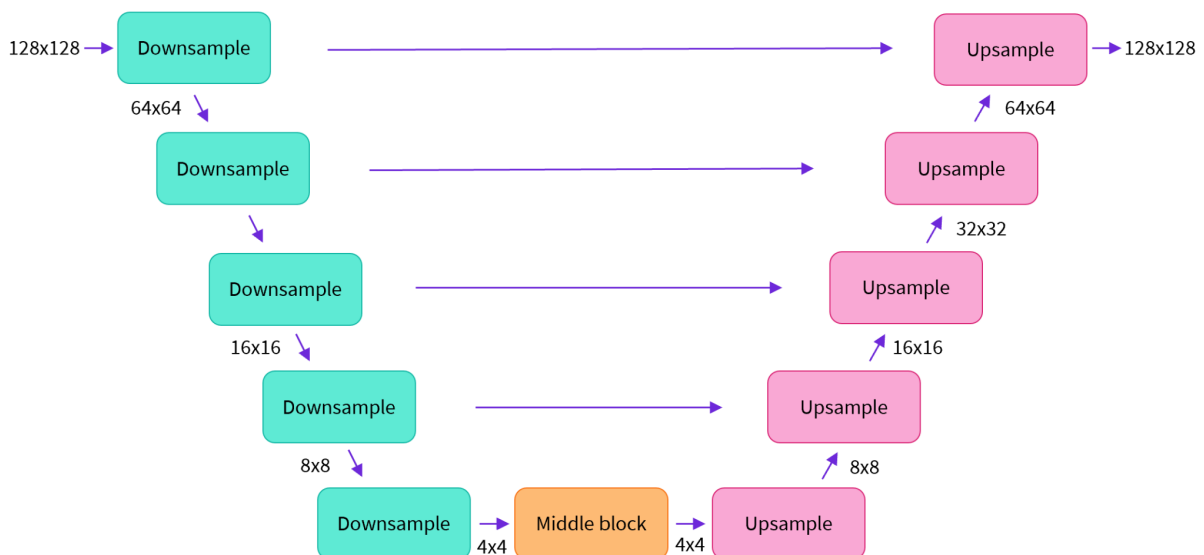
To understand “stable diffusion”, we must first understand “diffusion models”, which are machine learning systems trained to denoise random gaussian noise step by step to get to an image sample. The neural network is trained to predict a way to slightly denoise the image in each step. After a certain number of steps, a sample is obtained:



To train a diffusion model, a neural network learns to denoise data gradually starting from pure noise: First, a fixed forward diffusion process that gradually adds Gaussian noise to an image until you end up with pure noise. Secondly, a learned reverse denoising diffusion process p_θ , where a neural network is trained to gradually denoise an image from pure noise until you end up with an actual image. The second step is illustrated in the following image:



The model architecture consists of a U-Net model, where the model first downsamples the input (i.e. makes the input smaller in terms of spatial resolution), after which upsampling is performed:



To generate an image from given text input, the above model is conditioned on a domain-specific text encoder through the cross-attention mechanism. Thus we now have a model that takes text as input and outputs the desired related image.

Data

The model was trained on a large-scale dataset LAION-5B, a dataset of 5,85 billion CLIP-filtered image-text pairs, 14x bigger than LAION-400M, previously the biggest

openly accessible image-text dataset in the world. See this link for more info: <https://laion.ai/blog/laion-5b/>. The model was not trained to be factual or true representations of people or events, and therefore using the model to generate such content is out-of-scope for the abilities of this model. The limitation of this model includes 1) does not achieve perfect photorealism, 2) cannot render legible text, 3) faces and people may not be generated properly, 4) the model was trained mainly with English captions and will not work as well in other languages. The above information is also described here: <https://huggingface.co/CompVis/stable-diffusion-v1-4>.

Ethics

The model we are using is developed by: Robin Rombach and Patrick Esser. The model is released under the following license: The CreativeML OpenRAIL M license (<https://huggingface.co/spaces/CompVis/stable-diffusion-license>) is an Open RAIL M license (<https://www.licenses.ai/blog/2022/8/18/naming-convention-of-responsible-ai-license>), adapted from the work that BigScience and the RAIL Initiative are jointly carrying in the area of responsible AI licensing. See also the article about the BLOOM Open RAIL license on which our license is based: <https://bigscience.huggingface.co/blog/the-bigscience-rail-license>

Presentation Link

https://docs.google.com/presentation/d/127RI_HgLvcDDSjwp5LodmQ1Cf_pWXXWdemXjmmmmRRyA/edit?usp=sharing

GitHub link

<https://github.com/ywk991112/Stable-Diffusion-React-Web-App>

Code Folder/Organization

```
.
├── README.md (Steps to run the app)
├── app.py (Flask API)
├── diffusion.py (Diffusion Model)
├── src
│   ├── results (Storage of the generated images)
│   └── App.js (React Frontend)
```