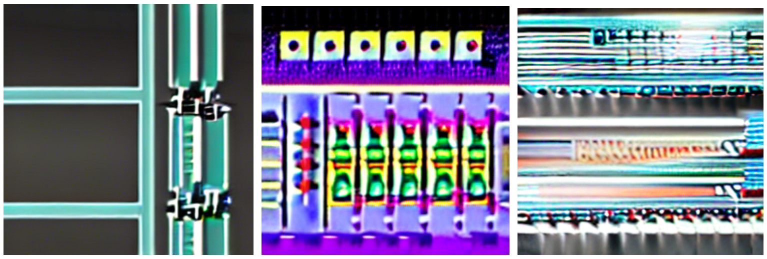
CS410 Project Progress Report

Team: VisualTopics
Team Members: Jason Lundquist (captain), Molly Yang, Louie Hernandez
Team Members NetID: jasondl3, tyy2, lgherna2

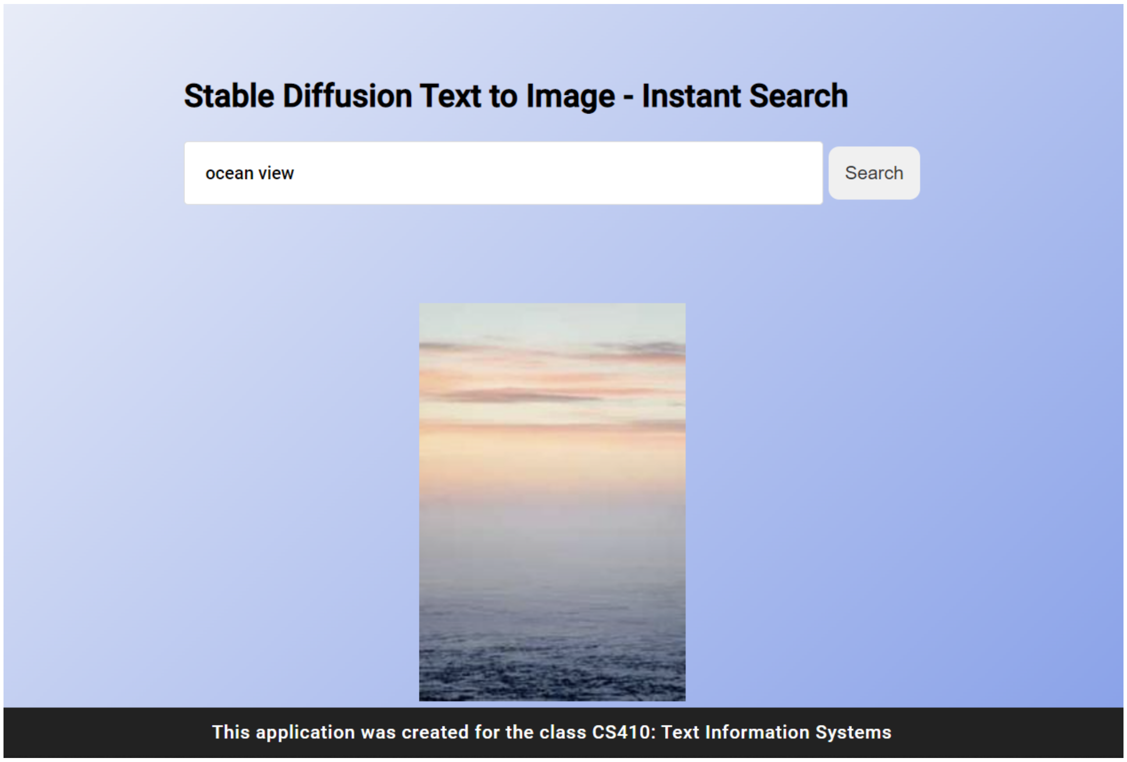
Completed Tasks

We have made progress with the survey of Stabile Diffusion, DALL·E 2 and Midjourney. Completed local Stabile Diffusion install and test run on Linux host, successfully creating baseline images using Stabile Diffusion checkpoint (Stable-Diffusion-v-1-4: https://huggingface.co/CompVis/stable-diffusion-v-1-4-original). The following images were generated with the prompt "electric circuit":



Researched prompt engineering tools for text-to-image AI generative models like Stable Diffusion, DALL·E 2 and Midjourney. There are prompt generators that use AI to build and enhance prompts (Promptly), or entering an image to generate a prompt. Additionally, evaluated different areas in academics where diagrams could be helpful for students to learn. The two areas we explored were structural diagrams and circuits diagrams. Both areas have the potential to be very simple and expand to increase complexity. The availability and quality of the datasets was considered when we decided to focus on a model that specializes in electronics. The sources of our data would include but not limited to Quizlet flashcards that other users have generated, online practice exams, electrical schematic software tools like Fritzing or Autodesk EAGLE.

The UI has been completed using a two tier structure: frontend written using VanillaJS and a backend written using Flask. Here is the MVP.



Stable Diffusion Text to Image - Instant Search

ocean view

This application was created for the class CS410: Text Information Systems

Lastly, conducted research on other machine learning models such as Dall-E, which was developed by the OpenAI group to generate digital images from natural language descriptions, ie: “prompts”. Once we are done with training we will compare results from the various prompts we come up with.

Pending Tasks

Pending tasks include creating a custom checkpoint build of electronic schema and component images, this effort includes creating and locating existing images sourced from public domain.

The Flask API still needs to be fleshed out, images are going to be sent to the frontend using a single endpoint. Once we fine tune the parameters for the text2img function we shall reference that in the API.

We also need to build and release a public facing environment to host the system for testing and final release.

Challenges

Model compute resources (require Nvidia graphics card > 4 GB) - needed to compromise on resolution. Instead of 512 x 512 images, we decreased the resolution to 256 x 256 to optimize runtime efficiency. Alternatively, Colab notebook includes sufficient VRAM for the images to be generated at full 512 x 512 resolution, but is more limiting for fine tuning and editing the model. To account for this we did research on forks of the stable diffusion project and stumbled upon this repo: https://github.com/basujindal/stable-diffusion.git. The tagline in the README is "Optimized Stable Diffusion", it is modified to use less VRAM than the original by sacrificing inference speed. To reduce the VRAM usage the following optimizations were done: the stable diffusion model is fragmented into four parts which are sent to the GPU only when needed. After the calculation is done, they are moved back to the CPU; the attention calculation is done in parts. The optimized code is found in the folder optimizedSD/ , when invoking the text2img file in that folder it follows the same parameters as the original copy found in the scripts/ folder. After invoking these optimized scripts and setting the dimensions to 256x256 I was able to use the model.