CS410 Project Proposal

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

Topics Description

State-of-the-art text to visualization applications exist today that bring together information retrieval and deep learning to create stunning images from merely a couple of words in the query. The models are trained on billions of text image pairs.[1] However, this extraordinary ability comes with certain limitations and societal impact. Some examples include the datasets (LAION, further details in the appendix) reflecting harmful stereotypes, oppressive viewpoints, and social bias. Efforts were made to filter out inappropriate content and toxic language; however, some like Imagen decided not to release the software for public use considering the risk of the model encoded harmful representations.[2]

Our goal for the final project is to explore a route to not only exclude the harmful content, but also apply this state-of-the-art technology to education. Many of the results we get from today's text to image models are utilized for a quick and fun way to conceptualize an idea, entertainment, or mimicking art styles. We wonder if the same technology could contribute to students and educators in classrooms. When it comes to studying for subjects like math, physics, or chemistry, having an image or diagram along a long paragraph of text could make a world of a difference. Like having subtitles to a video, we want to be able to perform the reversed task and come up with a visualization (image, slides of images, or even a video) to a paragraph.

We will begin with a thorough survey of the current technologies. When we have a final model we may compare results from descriptions with other text to image technologies like DALL-e 2. Comparisons may be subjective but it would be interesting to see different models interpretations from the input.

We will attempt to run a pretrained model (Stable Diffusion - see appendix) sentence by sentence in the paragraph. Then, explore ways to further filter the result and outputs. Another approach would be to find or generate a dataset with educational content for the subject of our choice to train the model. We will evaluate the results by level of relevancy rated by team members. In the end of the project, we will focus on pointing out the positives and negatives of the approaches we explored as well as suggestions for future development.

[1] https://laion.ai/blog/laion-5b/
[2] https://imagen.research.google/

Programming Languages

Python
Packages used:
Stable Diffusion has the diffusers library integrated into its codebase. Authors suggest downloading and sampling the library.
https://github.com/huggingface/diffusers.git

Workload

An estimated effort for the project:

Topic Modeling

Research: survey existing datasets, models, applications	20hrs
Exploration: based on initial results, come up with different approaches	20hrs
Implementation: Ubuntu 20 with 16 cores, 32GM RAM, NVIDIA 12GB RAM - 6 GPU	20hrs
Output Profiling	10hrs
Total	**70hrs**

Appendix

Stable Diffusion - Text-to-Image Generator

For our project we have chosen to pursue a free topic: Stable Diffusion text-to-image. Stable Diffusion is an image generator that can deliver breakthrough speed and quality while running on consumer GPUs. The model is based on the latent diffused model created by CompVis and Runway but enhanced with insights from conditional diffusion models by Stable Diffusion's lead generative AI developer Katherine Crowson, Open AI, Google Brain, and others. The core dataset has been trained on LAION-Aesthetics, a dataset that filters the 5.85 billion images in the **LAION-5B** dataset based on how "beautiful" an image was, building on ratings from the alpha testers of Stable Diffusion. Stable Diffusion runs on systems with under 10GB of VRAM and generates 512×512 pixel resolution images in just a few seconds. This model will take an input such as: `prompt = "a photograph of an astronaut riding a horse"`, and output an image as in the example below:

Stable Diffusion

Stable Diffusion is based on a particular type of diffusion model called Latent Diffusion, proposed in High-Resolution Image Synthesis with Latent Diffusion Models. General diffusion models are machine learning systems that are trained to denoise random gaussian noise step

by step, to get to a sample of interest, such as an image. Diffusion models have shown to achieve state-of-the-art results for generating image data. But one downside of diffusion models is that the reverse denoising process is slow. In addition, these models consume a lot of memory because they operate in pixel space, which becomes unreasonably expensive when generating high-resolution images. Therefore, it is challenging to train these models and also use them for inference.

Latent diffusion can reduce the memory and compute complexity by applying the diffusion process over a lower dimensional latent space, instead of using the actual pixel space. This is the key difference between standard diffusion and latent diffusion models: in latent diffusion the model is trained to generate latent (compressed) representations of the images.

There are three main components in latent diffusion.

An autoencoder (VAE) - The VAE model has two parts, an encoder and a decoder. The encoder is used to convert the image into a low dimensional latent representation, which will serve as the input to the U-Net model. The decoder, conversely, transforms the latent representation back into an image. During latent diffusion training, the encoder is used to get the latent representations (latents) of the images for the forward diffusion process, which applies more and more noise at each step. During inference, the denoised latents generated by the reverse diffusion process are converted back into images using the VAE decoder. As we will see during inference we only need the VAE decoder.

A U-Net - The U-Net has an encoder part and a decoder part both composed of ResNet blocks. The encoder compresses an image representation into a lower resolution image representation and the decoder decodes the lower resolution image representation back to the original higher resolution image representation that is supposedly less noisy. More specifically, the U-Net output predicts the noise residual which can be used to compute the predicted denoised image representation. To prevent the U-Net from losing important information while downsampling, short-cut connections are usually added between the downsampling ResNets of the encoder to the upsampling ResNets of the decoder. Additionally, the stable diffusion U-Net is able to condition its output on text-embeddings via cross-attention layers. The cross-attention layers are added to both the encoder and decoder part of the U-Net usually between ResNet blocks.

A text-encoder, e.g. CLIP's Text Encoder - The text-encoder is responsible for transforming the input prompt, e.g. "An astronaut riding a horse" into an embedding space that can be understood by the U-Net. It is usually a simple transformer-based encoder that maps a sequence of input tokens to a sequence of latent text-embeddings. Inspired by Imagen, Stable Diffusion does not train the text-encoder during training and simply uses CLIP's already trained text encoder, CLIPTextModel.

Why is latent diffusion fast and efficient?

Since the U-Net of latent diffusion models operates on a low dimensional space, it greatly reduces the memory and compute requirements compared to pixel-space diffusion models. For example, the autoencoder used in Stable Diffusion has a reduction factor of 8. This means that an image of shape (3, 512, 512) becomes (3, 64, 64) in latent space, which requires 8 × 8 = 64 times less memory. This is why it's possible to generate 512 × 512 images so quickly, even on 16GB Colab GPUs!

Stable Diffusion during inference

Logical flow: