REPORT

GENERATIVE AI (CONTROL NET+STABLE DIFFUSION)

SHRISHTI SHAH
SHUBHASRI TADEPALLI
LALITHA TANMAI VADDIPARTHI

GENERATIVE AI

Generative AI is a type of artificial intelligence technology that can produce various types of content, including text, imagery, audio and synthetic data. The recent buzz around generative AI has been driven by the simplicity of new user interfaces for creating high-quality text, graphics and videos in a matter of seconds.

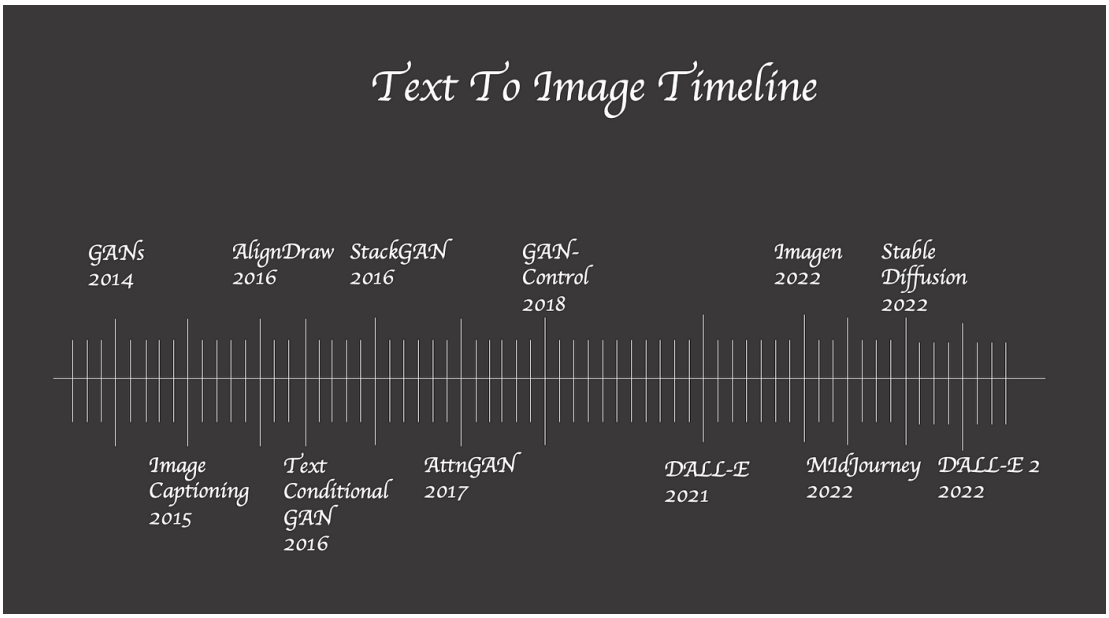
The technology, it should be noted, is not brand-new. Generative AI was introduced in the 1960s in chatbots. But it was not until 2014, with the introduction of generative adversarial networks, or GANs -- a type of machine learning algorithm -- that generative AI could create convincingly authentic images, videos and audio of real people.

HOW DOES GENERATIVE AI WORK

Generative AI starts with a prompt that could be in the form of a text, an image, a video, a design, musical notes, or any input that the AI system can process. Various AI algorithms then return new content in response to the prompt. Content can include essays, solutions to problems, or realistic fakes created from pictures or audio of a person.

Early versions of generative AI required submitting data via an API or an otherwise complicated process. Developers had to familiarize themselves with special tools and write applications using languages such as Python.

One of the primary advantages of generative AI in content marketing is its ability to create content at scale. With the increasing demand for content, it can be challenging for businesses to produce quality content consistently



FEW GAN ARCHITECTURES

Vanilla GAN

Vanilla GAN is a term often used to refer to the original and basic form of a Generative Adversarial Network (GAN).There are 2 kinds of models in the context of Supervised Learning, Generative and Discriminative Models.

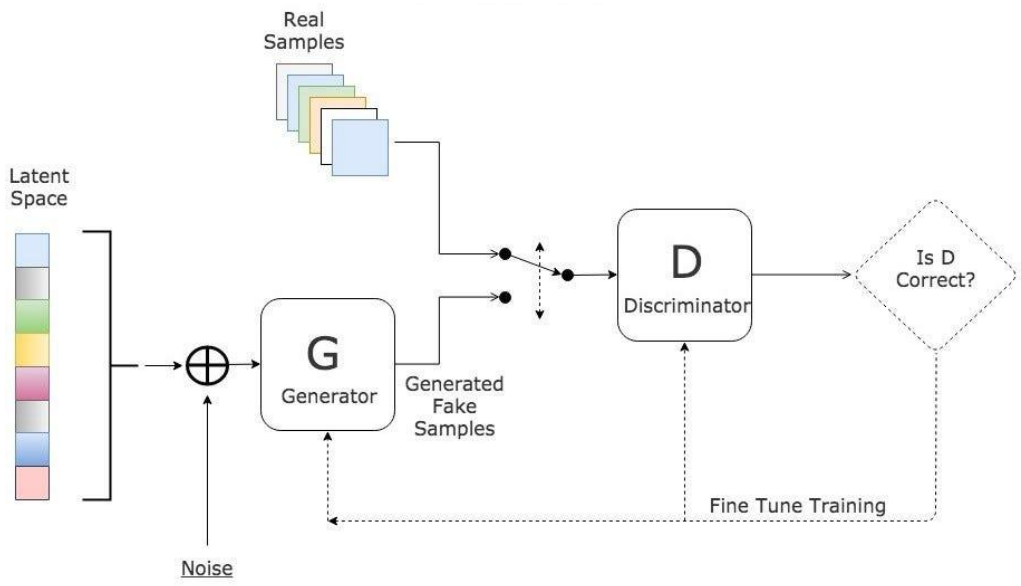
Goal:
The goal of Vanilla GAN in general is to generate new images via learning from the original image data. The model primarily consists of two parts, i.e. a generator which produces image from noise z and a discriminator. which tries to distinguish between z and the true images x.

Generator : The generator's primary task is to create synthetic data that resembles real data. It takes random noise as input and generates data samples. The generator is typically implemented as a neural network that maps random noise vectors to data samples.

Discriminator : The discriminator's role is to distinguish between real data and fake data generated by the generator. It also takes the form of a neural network,

but instead of generating data, it takes data samples as input and outputs a probability score indicating whether the input data is real or fake.

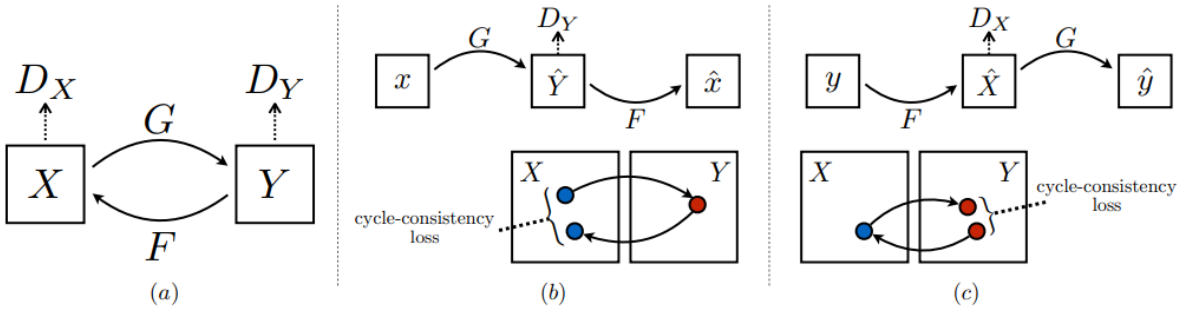
ARCHITECTURE



Cycle GAN

The Cycle Generative Adversarial Network, or CycleGAN is a type of generative adversarial network used for image-to-image translation tasks.The main idea behind CycleGAN is to learn a mapping between two different domain without requiring paired training data.

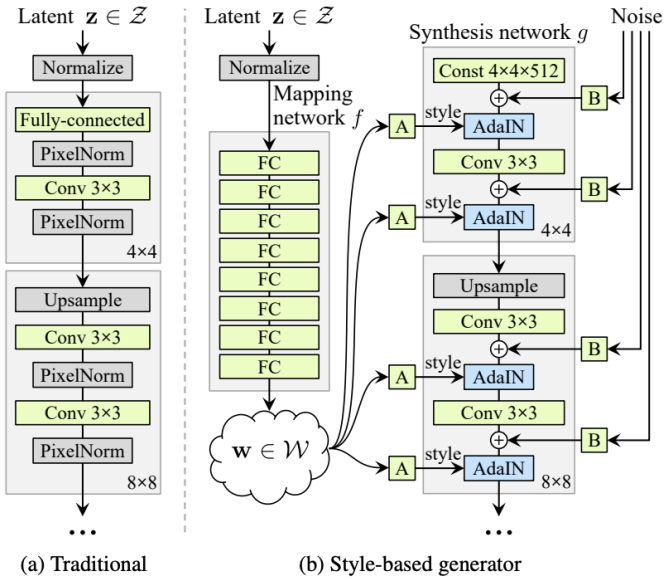
CycleGAN is an approach to training a deep convolutional neural network for image-to-image translation tasks.
CycleGAN's biggest advantage is that it can produce remarkable results without the need for a paired dataset. Moreover, it works well for texture and color changes



Style GAN

For generating high-quality images with controllable styles and attributes.Initially latent vector was normalized using PixelNorm.High quality dataset namely Flickr Faces- HQ (FFHQ).

StyleGAN is a type of generative adversarial network. It uses an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature; in particular, the use of adaptive instance normalization.
One of the notable advantages of StyleGAN is its ability to generate highly realistic and diverse images, making it a valuable tool for content creation and visual design. However, StyleGAN's reliance on large-scale training data and computational resources can pose challenges for some applications.



StackGAN

StackGAN, which stands for "Stacked Generative Adversarial Networks," is a generative model architecture designed for text-to-image synthesis. It is specifically developed to generate high-resolution images from textual descriptions.

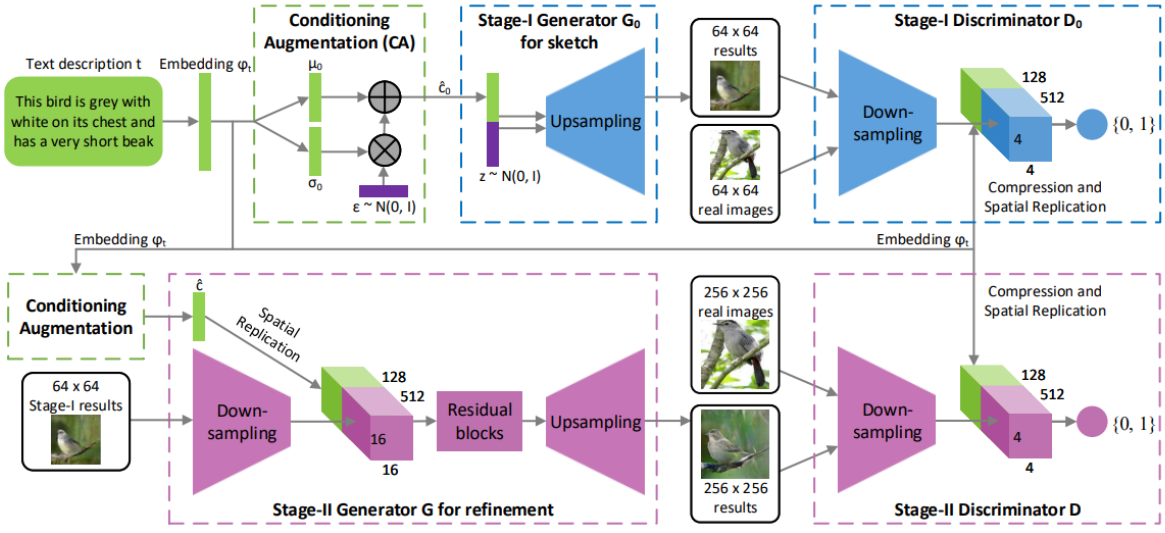
To generate high-resolution images with photo-realistic details, we propose a simple yet effective StackedGAN. It decomposes the text-to-image generative process into two stages :

Stage-I GAN: it sketches the primitive shape and basic colors of the object conditioned on the given text description, and draws the background layout from a random noise vector, yielding a low-resolution image.

Stage-II GAN: it corrects defects in the low-resolution image from Stage-I and completes details of the object by reading the text description again, producing a high resolution photo-realistic image.

Practical applications of StackGAN

- Generating high-resolution images automatically for entertainment purposes or educational purposes.
- Creating comics: With the use of a StackGAN, the process of creating comics can be reduced to days as the StackGAN can generate comics automatically and assist in the creative process.



Dataset :

CUB Dataset - birds Oxford-102 Dataset - flowers MS COCO Dataset - backgrounds and multiple objects

Dall-E2

(Text-conditional Image generation)

DALL-E 2 is a state-of-the-art AI-powered image-generating platform created by OpenAI. It is the advanced and improved successor of DALL-E. It can produce incredibly realistic and detailed pictures based on textual descriptions by utilizing deep learning techniques.

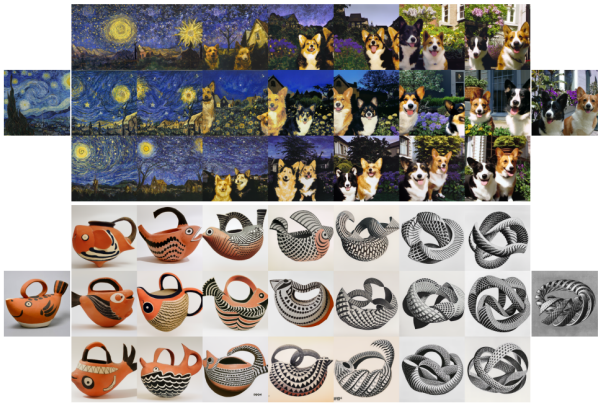
DALL-E 2 is a state-of-the-art AI-powered image-generating platform created by OpenAI. It is the advanced and improved successor of DALL-E. It can produce incredibly realistic and detailed pictures based on textual descriptions by utilizing deep learning techniques

Uses of Dall-E

1. Can create original realistic images and art from text description.it can combine concepts,attributes and styles.
2. Generating image variations.

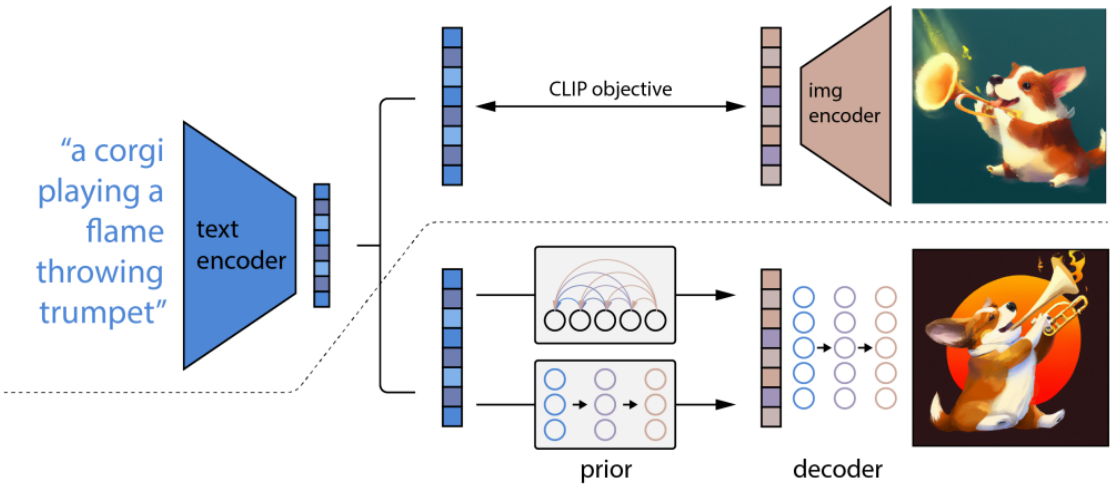
3. Language guided image manipulations

4. Generating image interpolations.



5.Can make realistic edits to existing images from a natural language caption.It can add and remove elements while taking shadows ,reflections and textures into account.

Dall-E2 Architecture



Architecture of Dall-E2 In this work, we combine these two approaches for the problem of text-conditional image generation. We first train a diffusion decoder to invert the CLIP image encoder. Our inverter is non-deterministic, and can produce multiple images corresponding to a given image embedding.. Training Stage (Above the Dotted Line):

● CLIP Training Process: In this phase, the model undergoes training to create a joint representation space for both text and images. This means that CLIP learns to associate text descriptions with corresponding images, essentially understanding the relationships between them. Text-to-Image Generation Stage (Below the Dotted Line):

● Text Input: The process of generating images from text captions begins by providing a text caption as input.

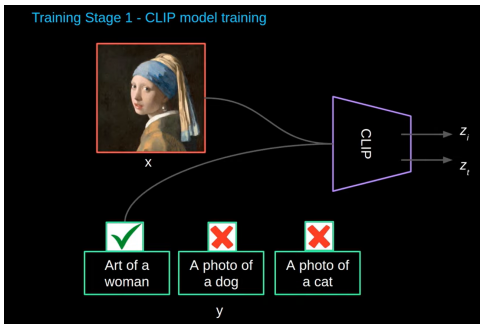
● CLIP Text Embedding: The input text caption is converted into a CLIP text embedding. This embedding captures the meaning and style of the text description. Importantly, the CLIP model, which is pretrained and has learned this shared representation space, is used to perform this conversion.

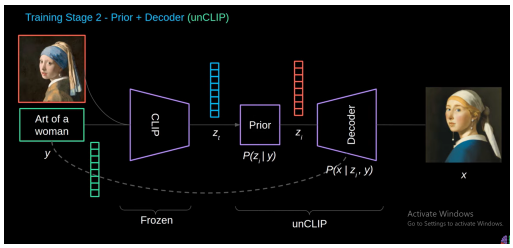
● Prior Model (Autoregressive or Diffusion): The CLIP text embedding is then fed into a prior model. This prior model can either be autoregressive or diffusion-based. The prior's role is to generate an image embedding based on the CLIP text embedding. This image embedding is a numerical representation that encodes the information needed to create an image that aligns with the text's content and style.

● Image Embedding: The prior successfully produces the image embedding, which represents the desired image content and style guided by the input text.

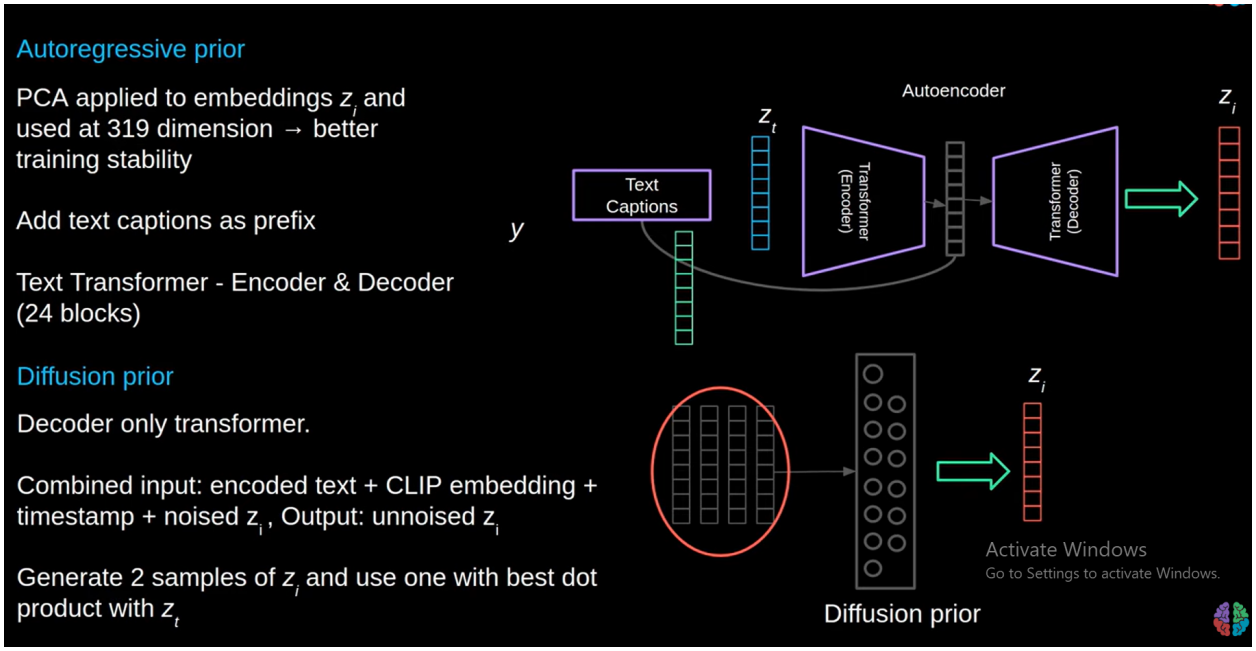
● Diffusion Decoder: The image embedding is used to condition a diffusion decoder. The diffusion decoder is responsible for taking this embedding and transforming it into the final image. The decoder generates pixel-level details and overall image structure based on the information contained in the image embedding.

STAGES

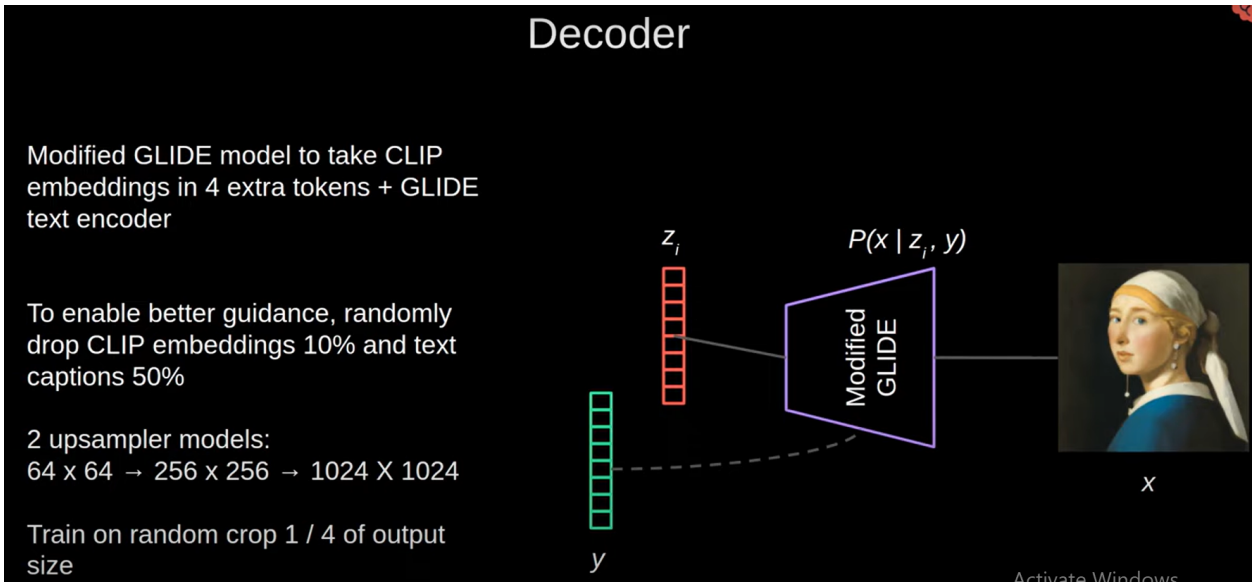




Prior Model (Autoregressive or Diffusion): The CLIP text embedding is then fed into a prior model. This prior model can either be autoregressive or diffusion-based. The prior's role is to generate an image embedding based on the CLIP text embedding. This image embedding is a numerical representation that encodes the information needed to create an image that aligns with the text's content and style.



Diffusion Decoder: The image embedding is used to condition a diffusion decoder. The diffusion decoder is responsible for taking this embedding and transforming it into the final image. The decoder generates pixel-level details and overall image structure based on the information contained in the image embedding.



Importance of prior :

Human Evaluation

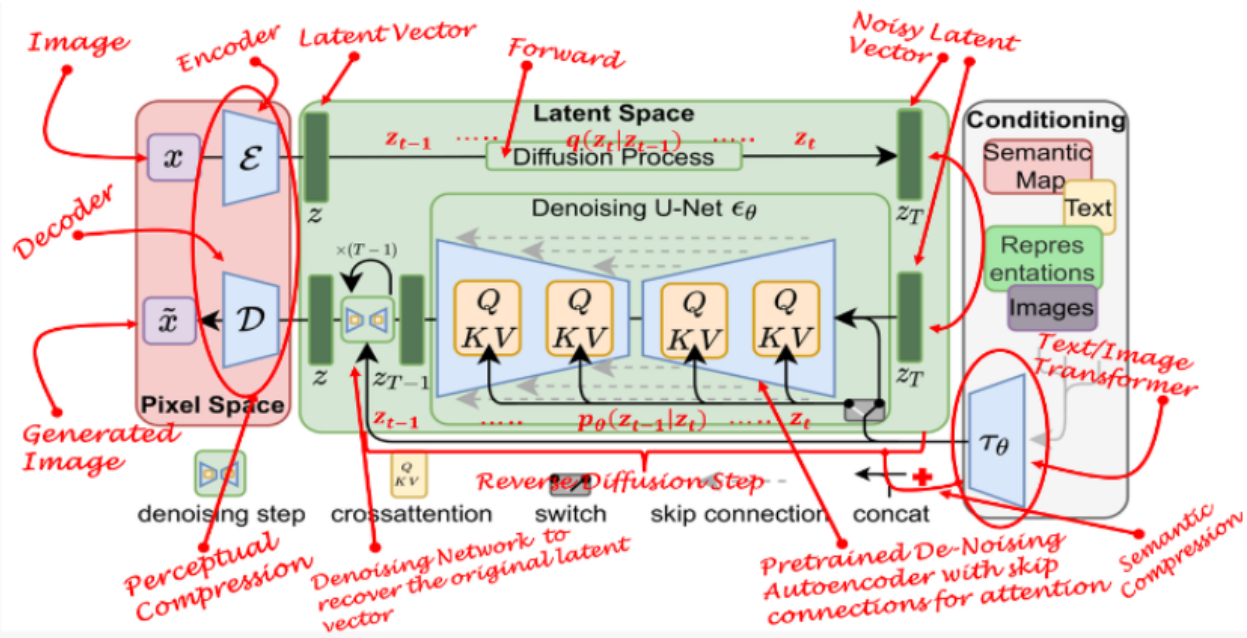
unCLIP Prior	Photorealism	Caption Similarity	Diversity
AR	$47.1\% \pm 3.1\%$	$41.1\% \pm 3.0\%$	$62.6\% \pm 3.0\%$
Diffusion	$48.9\% \pm 3.1\%$	$45.3\% \pm 3.0\%$	$70.5\% \pm 2.8\%$

Model	FID	Zero-shot FID	Zero-shot FID (filt)
AttnGAN (Xu et al., 2017)	35.49		
DM-GAN (Zhu et al., 2019)	32.64		
DF-GAN (Tao et al., 2020)	21.42		
DM-GAN + CL (Ye et al., 2021)	20.79		
XMC-GAN (Zhang et al., 2021)	9.33		
LAFITE (Zhou et al., 2021)	8.12		
Make-A-Scene (Gafni et al., 2022)	**7.55**		
DALL-E (Ramesh et al., 2021)		~ 28	
LAFITE (Zhou et al., 2021)		26.94	
GLIDE (Nichol et al., 2021)		12.24	12.89
Make-A-Scene (Gafni et al., 2022)			11.84
unCLIP (AR prior)		10.63	11.08
unCLIP (Diffusion prior)		**10.39**	**10.87**

Frechet Inception Distance (FID) FID measures the similarity between the distribution of real images and generated images in feature space, using the Inception model's activations. Lower FID scores are indicative of better quality and diversity.

Stable diffusion

Stable Diffusion is a latent text-to-image diffusion model capable of generating photo-realistic images given any text input.Stable Diffusion is a text-to-image model created by a collaboration between engineers and researchers from CompVis, Stability AI, and LAION.



The Variational Autoencoder (VAE) neural network has two parts: (1) an encoder and (2) a decoder. The encoder compresses an image to a lower dimensional representation in the latent space. The decoder restores the image from the latent space.

The latent space of the Stable Diffusion model is 4x64x64, 48 times smaller than the image pixel space. All the forward and reverse diffusions we talked about are actually done in the latent space. So during training, instead of generating a noisy image, it generates a random tensor in latent space (latent noise). Instead of corrupting an image with noise, it corrupts the representation of the image in latent space with the latent noise. The reason for doing that is it is a lot faster since the latent space is smaller.

ControlNet

ControlNet is a neural network that controls a pretrained image Diffusion model (e.g. Stable Diffusion). Its function is to allow input of a conditioning image, which can then be used to manipulate the image generation.

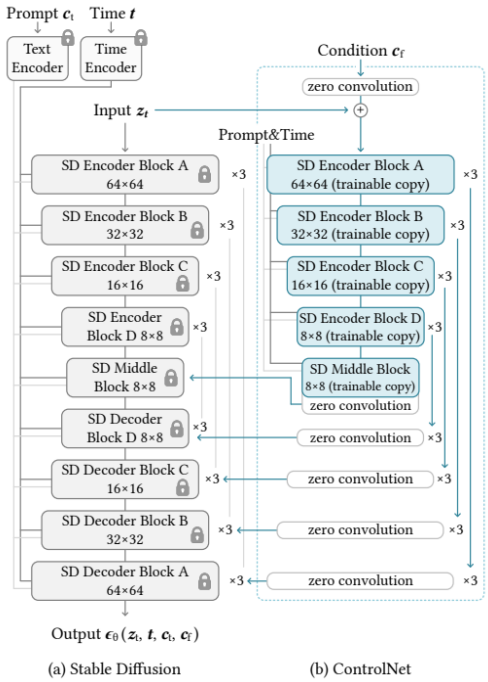
What Does ControlNet Do?

The combination of ControlNet and Stable Diffusion enables Stable Diffusion to take in a condition input that guides the image generation process, resulting in enhanced performance of Stable Diffusion.

It can accept scribbles, edge maps, pose key points, depth maps, segmentation maps, normal maps, etc as the condition input to guide the content of the generated image.

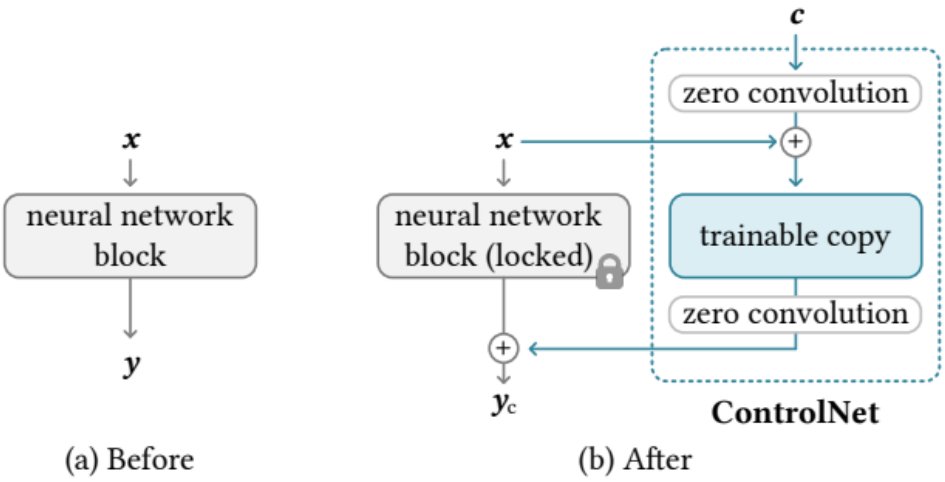
ControlNet is a neural network structure to control diffusion models by adding extra conditions.It copies the weights of neural network blocks into a "locked" copy and a "trainable" copy.The "trainable" one learns your condition. The "locked" one preserves your model.

ARCHITECTURE

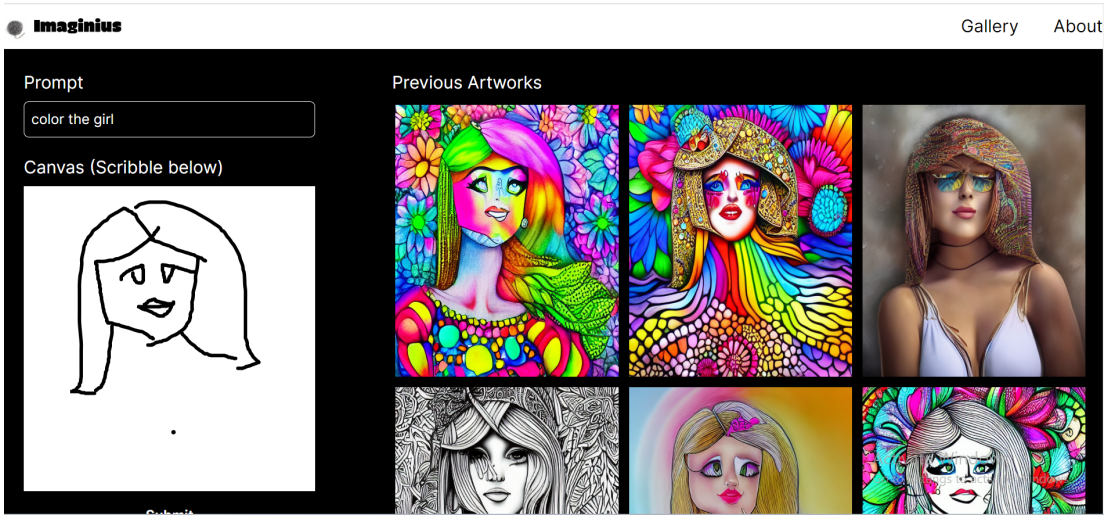


It copies the weights of neural network blocks into a "locked" copy and a "trainable" copy.The "trainable" one learns your condition. The "locked" one preserves your

model.Training with a small dataset of image pairs will not destroy the production-ready diffusion models. The "zero convolution" is 1×1 convolution with both weight and bias initialized as zeros. Before training, all zero convolutions output zeros, and ControlNet will not cause any distortion. No layer is trained from scratch. You are still fine-tuning. Your original model is safe.



We have used ControlNet + Stable Diffusion with Scribble



Model Performance

To assess the quality of images created by generative models, it is common to use the Fréchet inception distance (FID) metric. In a nutshell, FID calculates the distance between the feature vectors of real images and generated images. On the COCO benchmark, Imagen currently achieved the best (lowest) zero-shot FID score of 7.27, outperforming DALL·E 2 with a 10.39 FID score.

What is the inception score?

The inception score (IS) is a mathematical algorithm used to measure or determine the quality of images created by generative AI through a generative adversarial network (GAN). The word "inception" refers to the spark of creativity or initial beginning of a thought or action traditionally experienced by humans.

Tech stack

Welcome to our innovative generative AI website, where creativity knows no bounds! Transform your scribbles into realistic and vibrant representations with the power of cutting-edge technology. Crafted using Next.js for a seamless and dynamic user experience, and styled with the sleek aesthetics of Tailwind CSS. Behind the scenes, our backend and database are powered by Convex, ensuring robust performance and data management. The magic happens through the art of machine learning, facilitated by Replicate.com(https://replicate.com/jagilley/controlnet-scribble), bringing your imagination to life in vivid colors and stunning realism.

Replicate API is built on the Paper (https://arxiv.org/abs/2302.05543)Adding Conditional Control to Text-to-Image Diffusion Models , the model. This model runs on Nvidia A100 (40GB) GPU hardware. Predictions typically complete within 7 seconds.

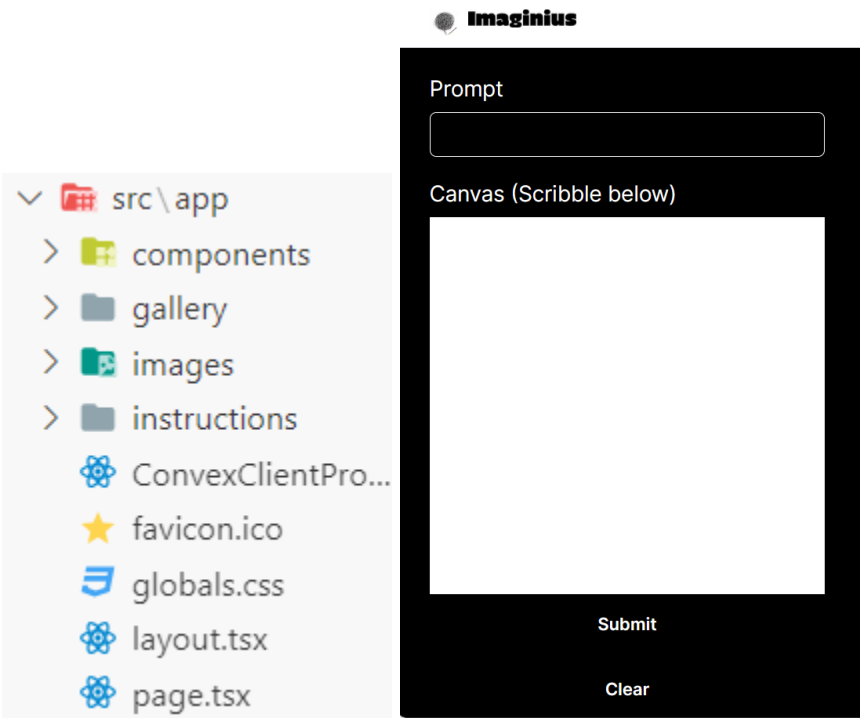
How to run the website

Frontend-
Node.js, Tailwind CSS

All the dependencies entioned below are required

```
"dependencies": {
    "@types/node": "20.4.2",
    "@types/react": "18.2.15",
    "@types/react-dom": "18.2.7",
    "convex": "^0.19.1",
    "eslint": "8.45.0",
    "eslint-config-next": "13.4.10",
    "next": "^13.4.10",
    "react": "18.2.0",
    "react-dom": "18.2.0",
    "react-hook-form": "^7.47.0",
    "react-sketch-canvas": "^6.2.0",
    "replicate": "^0.12.3",
    "typescript": "5.1.6"
  },
```

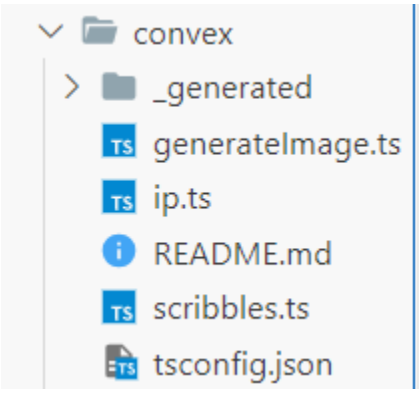
Command to create the application -npx create-next-app@latest imaginius
Command to build - npm run build
Command to run-npm start



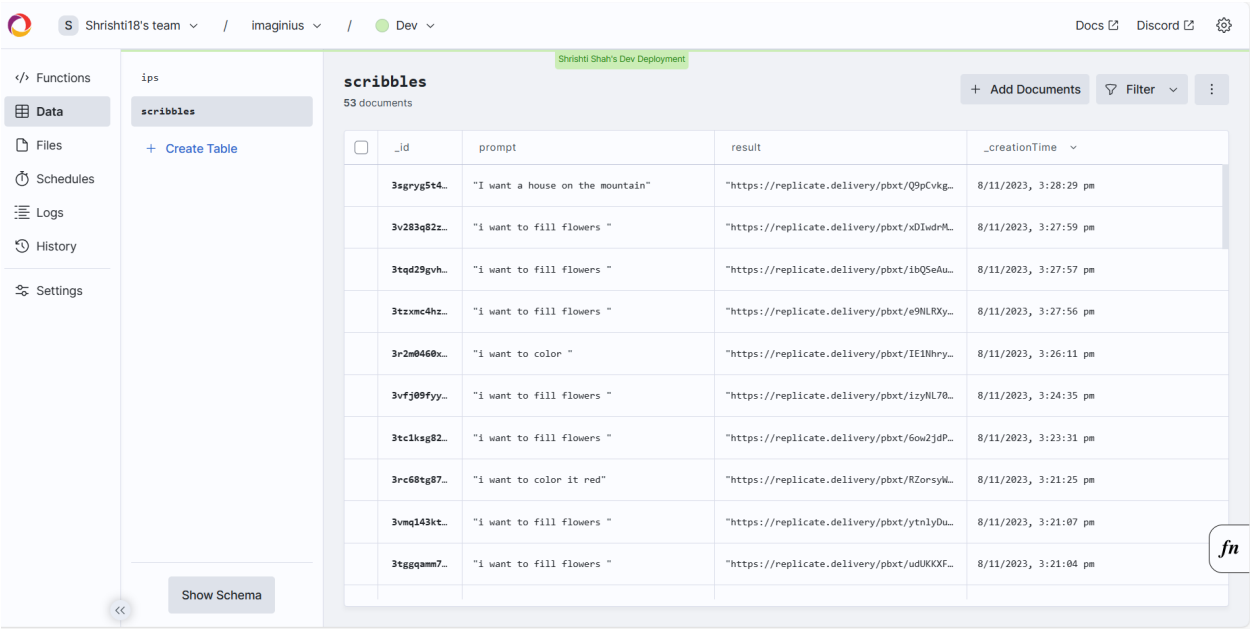
This is the structure of our frontend files.

Page.tsx is the start file which connects all the respective files as their respective needs. Its takes input of the promt , the scribble on canva and as you submit puts in the backend.

ConvexClientProvider is a connection of our frontend with the backend which is in Convex.



These are the backend files, in which
The prompt and scribble which we have sent in the table of backend, is extracted and using replicate API is converted into a scribble with new generated colorful images according to the prompt.



The backend data you can see _id, prompt, result and creation Time.

8/11/2023, 3:28:29 pm	18ms	success	Query	scribbles:getScribbles
8/11/2023, 3:28:33 pm	13ms	success	Mutation	scribbles:updateScribbleResult
8/11/2023, 3:28:33 pm	17ms	success	Query	scribbles:getScribbles
8/11/2023, 3:28:29 pm	4071ms	success	Action	generateImage:generateImage
8/11/2023, 6:42:56 pm	33ms	success	Query	scribbles:getScribbles
10/11/2023, 11:48:47 am	S shrishtishah2002@gmail.com updated the configuration			
10/11/2023, 11:50:43 am	22ms	success	Query	scribbles:getScribbles
10/11/2023, 11:50:43 am	13ms	success	Mutation	ip:saveIpAddress
10/11/2023, 11:50:43 am	CACHED	success	Query	scribbles:getScribbles

Backend Functions.