Dissertation Submitted for the partial fulfillment of the **M.Sc. (Integrated) Data Science** degree to the Department of AI&ML and Data Science

M.Sc. Project Dissertation

Semester- X

StyleGEN

submitted to

By

Jay Soni

Under the guidance of

Meet Yogi

M.Sc. (Integrated) Data Science

Department of AIML & Data Science

School of Emerging Science and Technology

Gujarat University

April, 2024

DECLARATION

I hereby declare that the study entitled **StyleGen** submitted to the Gujarat University, Navarangpura, Ahmedabad (Gujarat) in partial fulfilment of M.Sc. (Int) Data Science degree is the result of investigation done by myself. The material that has been obtained (and used) from other sources has been duly acknowledged in this study. It has not been previously submitted either in part or whole to this or any other university or institution for the award of any degree or diploma.

Place: Ahmedabad
Date: 29-04-24

Signature
Jay Soni

Index

1. Abstract & Key Words

StyleGen: Text-to-Image Generation with Stable Diffusion

The ever-evolving landscape of text-to-image generation models presents exciting possibilities for creative expression. This report delves into styleGen, a project that harnesses the power of different **Stable Diffusion** models and empowers users with artistic control over the generated imagery.

At the core of styleGen lies a custom **LoRA** (Low Rank Adaption) model. This model, meticulously trained on Custom datasets with base of Stable Diffusion model, allows for the targeted incorporation of specific artistic styles into the generated images. Web Extracted custom data is around 100-150 pic after preprossing on data only 30-35 images are used for training, Users can leverage text prompts that guide the overall composition while simultaneously influencing the final image with a desired generated image flair.

StyleGen extends its functionality beyond basic text prompts by incorporating a **ControlNet** model. This grants users a layer of granular control over the generation process. They can utilize reference images or manipulate specific aspects like depth and composition, further refining the artistic direction of the final output.

Finally, styleGen integrates a **refiner model** specifically tailored for Stable Diffusion. This model acts as a finishing touch, enhancing details and ensuring a polished final image.

Model fine-tuned on stable diffusion are deployed on a open source platform called Hugging Face. Where they provide serverless deployment for custom deployed models.

Furthermore, styleGen is presented as a user-friendly **Streamlit** web application, making it readily accessible for creators of all technical backgrounds. This eliminates the need for complex coding knowledge and streamlines the image generation process.

This report comprehensively explores styleGen's functionalities. We delve into the methodology employed for training the LoRA model, analyze the

effectiveness of style transfer, and evaluate the control options offered by ControlNet. Additionally, the report assesses the impact of the refiner model on final image quality.

Keywords :

Stable Diffusion : A powerful latent diffusion model for generating high-resolution images from text descriptions.

LoRA (Low Rank Adaption) : Technique for fine-tuning diffusion models to incorporate specific artistic styles.

ControlNet model : Enables control over specific aspects of a generated image during the text-to-image process.

refiner model : Potentially an alternative model for making adjustments to the generated image after the initial creation process.

Hugging Face : A popular platform for sharing and using machine learning models, including Stable Diffusion and potentially future LoRA and ControlNet models.

Streamlit : A framework for creating user-friendly web applications, potentially used to deploy styleGen.

web application : A software program accessible through a web browser, making styleGen available to a wider audience.

User-friendly interface : Design elements that make styleGen easy to learn and use for people with varying technical backgrounds.

2. Introduction

This project explores the captivating world of text-to-image generation, leveraging the power of latent diffusion models like Stable Diffusion. Our main objective is to delve into the artistic possibilities this technology offers, empowering users with control over the creative process.

Introducing styleGen, a project built on Stable Diffusion that goes beyond basic text-to-image conversion. Through a custom LoRA model trained on specific datasets, styleGen allows users to infuse their chosen artistic style into generated images. Imagine conjuring a scene with the brushstrokes of Van Gogh or the pop-art vibrancy of Andy Warhol – styleGen empowers this artistic manipulation.

For even greater control, styleGen integrates a ControlNet model. This grants users the ability to modify specific visual elements within the generated image, independent of the text prompt or artistic style. This might involve adjusting a character's pose or refining details – ensuring the final output aligns with your artistic vision.

A dedicated refiner model further polishes the generated image, eliminating unwanted artifacts and enhancing details. This report will delve into the methodologies behind styleGen, analyzing the effectiveness of each component and its impact on artistic control and final image quality.

Finally, styleGen prioritizes user experience. Presented as a user-friendly Streamlit web application, it eliminates the need for complex coding, allowing creators of all backgrounds to explore their artistic vision with ease.

By enabling the creation of unique and stylistically defined images, complete with user-defined control, styleGen empowers users to push creative boundaries and explore uncharted territories. This project opens new doors for artistic exploration, paving the way for a future where technology seamlessly blends with human creativity.

3. Basic Terminology

1. **Latent Diffusion Model:** A type of deep learning model used for text-to-image generation. It works by iteratively refining a latent representation of an image based on text prompts. Stable Diffusion is a specific implementation of a latent diffusion model.

2. **Stable Diffusion:** A powerful latent diffusion model capable of generating high-resolution images from text descriptions. We explores the use of styleGen with a focus on Stable Diffusion XL 1.0, along with comparisons to earlier versions:

3. **Stable Diffusion XL 1.0 (SDXL 1.0):** The primary implementation used in styleGen. This stable diffusion model works better for image generation, it works well with high-resolution image generation then previous models.

4. **Stable Diffusion 1.5:** An earlier version of Stable Diffusion used for comparison. Our findings suggest that SDXL 1.0 performs better for StyleGen's purposes

5. **LoRA (Low-Rank Adaptation):** A technique used to fine-tune Stable Diffusion models towards specific artistic styles. We have used **custom LoRA model** trained on custom datasets, for effective image generation.

6. **Control-Net:** A Control-Net model is integrated for user control over aspects like pose, potentially independent of the chosen artistic style and primarily evaluated with SDXL 1.0, We have used open-pose Control net from diffusor.

7. **Refiner Model:** A refiner model specifically tailored for Stable Diffusion is used to enhance details in the final image, primarily evaluated with Lora image. But it is compatible with different versions of stable diffusion

4. Literature review

- styleGen pushes the boundaries of text-to-image generation by empowering users with artistic control. This innovative approach leverages advancements in three key areas. First, latent diffusion models, like Stable Diffusion [1], translate textual descriptions into high-resolution images, forming the core engine for styleGen. Second, styleGen builds upon artistic style transfer techniques. Traditionally complex, these techniques have benefited from deep learning advancements. The seminal work by Gatys et al. (2015) on convolutional neural networks for image style transfer paved the way for further research [2]. styleGen incorporates a custom LoRA (Low-Rank Adaptation) model, trained on specific datasets, allowing users to infuse their chosen artistic style into the generated image. Finally, research by Srinivasan et al. (2022) on ControlNet empowers user control over specific aspects of image generation [3]. styleGen integrates ControlNet, granting users the ability to define the pose of a figure within the generated image, independent of the chosen clothing style. By combining these advancements, styleGen unlocks a new frontier for artistic exploration, allowing users to not only create images from text but also infuse them with a desired generative style and control specific visual elements.

5. Methodology

1. Data :

1.1) Data Collection :

- Scraped data from different clothing e-commerce web site like : (Myntra, Google Images, and Ajio-fashion,Zara), Around 100-150 Images are collected for fine-tune diffusion model.

1.2) Preprocessing Image:

- Cleaning data by removing image whose quality is lower that 1080X1080. By doing this we will get high-resolution images after that removed blured or images with extra object for better training because extra object will may degrade the generation quality or hinder the fine tuning process and will get bad resulting.

1.3) Resizing image:

- To fine-tune stable diffusion model we need specific dimensions of the input image like for example : stable Diffusion XL 1.0 requires image of 512X512 , Stable Diffusion 1.5 model requires input image if size 256X256 or 512X512 but not lower than that
- To resize all image I have used tool called Birme which is used to resize image with main object in focus

After the processing done on data we will use that to fine-tune LoRa on this data. Here are some examples of data.

(1)

(2)

(3)

(4)

(5)

(6)

In LoRA for better fine-tune the model we have to add input Prompt in a string data type.
Here are the **examples** of the prompts for above Image.

(1) a women wearing a brown long sleeve tshirt and brown trouser, black pointed shoes, aligned to wall, with beige surface

(2) a women wearing a green long sleeve tshirt and blue denim pant, black and white converse shoes, potrait iamge, grey wall background

(3) a man wearing a white rolled sleeve and black pant, with white shoes, standing, potrait, offwhite background

(4) a women wearing a checked red and blue long sleeve tshirt, blue jeans, white shocks, beige and blue loafer shoes,

(5) a man wearing a brown turtle neck brown tshirt layerd with brown suit jacket, brown pant, black sunglasses, off white background

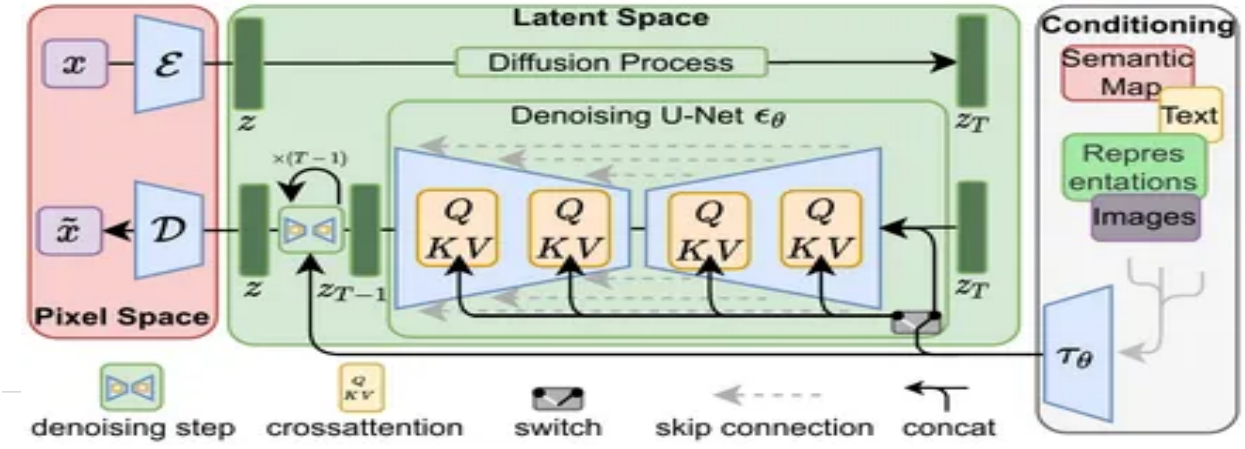
(6) a man wearing a white tshirt layered with orange shirt jacket, olive green pant, white sneakers, white background

2. Models :

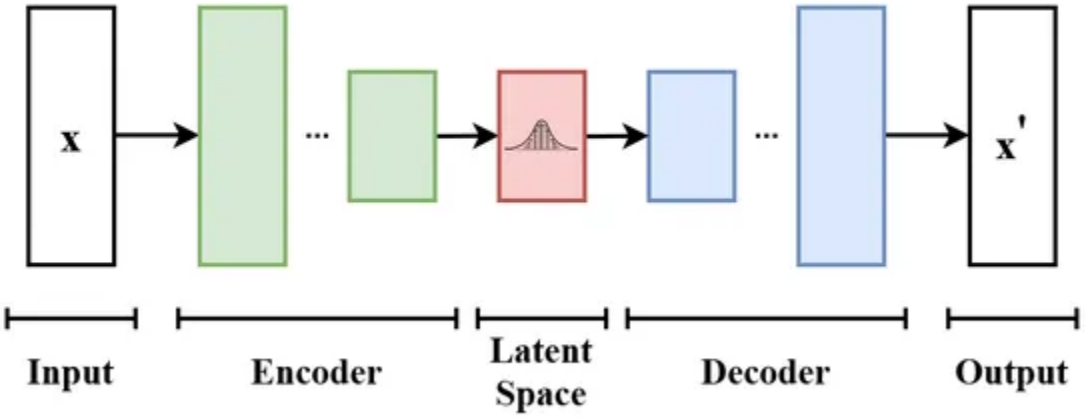
2.1) Stable Diffusion model :

Stable Diffusion models function as latent diffusion models. It learns the latent structure of input by modeling how the data attributes diffuse through the latent space. They belong to the deep generative neural network. It is considered stable because we guide the results using original images, text, etc. On the other hand, an unstable diffusion will be unpredictable.

Stable Diffusion uses the Diffusion or latent diffusion model (LDM), a probabilistic model. These models are trained like other deep learning models. Still, the objective here is removing the need for continuous applications of signal processing denoting a kind of noise in the signals in which the probability density function equals the normal distribution. We refer to this as the Gaussian noise applied to the training images. We achieve this through a sequence of denoising autoencoders (DAE).



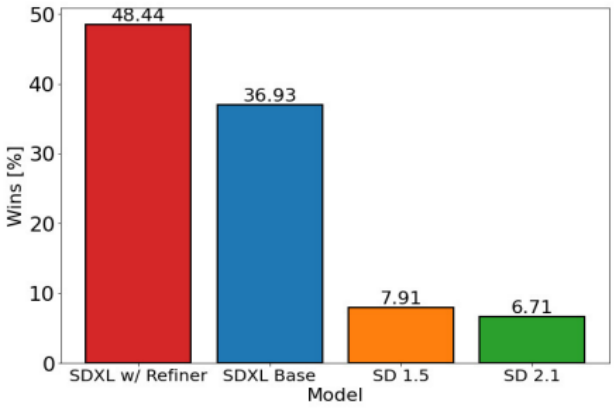
In a more detailed explanation, Stable Diffusion consists of 3 essential parts: First is the variational autoencoder (VAE) which, in simple terms, is an artificial neural network that performs as probabilistic graphical models. Next is the U-Net block. This convolutional neural network (CNN) was developed for image segmentation. Lastly is the text encoder part. A trained CLIP ViT-L/14 text encoder deals with this. It handles the transformations of the text prompts into an embedding space.



The VAE encoder compresses the image pixel space values into a smaller dimensional latent space to carry out image diffusion. This helps the image not to lose details. It is represented again in pixeled pictures.

There are multiple model of stable diffusion like
stable diffusion 1.5,
stable difffusion 2.1,
stable diffusion XL base,
stable diffusion XL base + refiner model
and many more variations.

According to Stability AI's own study, most users prefer the images from the SDXL model over the v1.5 base model. You will find a series of images generated with the same prompts from the v1.5 and SDXL models.



Differences between SDXL and v1.5 models :

The SDXL model is, in practice, two models. You run the base model, followed by the refiner model. The base model sets the global composition. The refiner model adds finer details. (You can optionally run the base model alone.)

The language model (the module that understands your prompts) is a combination of the largest OpenClip model (ViT-G/14) and OpenAI's proprietary CLIP ViT-L. Which means The prompts that work on v1.5 will have a good chance to work on SDXL.

The **U-Net**, the most crucial part of the diffusion model, is now 3 times larger. Together with the larger language model, the SDXL model generates high-quality images matching the prompt closely. The default image size of SDXL is 1024×1024. This is 4 times larger than v1.5 model's 512×512.

Example of output Image from both stable diffusion model 1.5 VS SD XL base:

Prompt : photo of young Caucasian woman, highlight hair, sitting outside restaurant, wearing dress, rim lighting, studio lighting, looking at the camera, dslr, ultra quality, sharp focus, Fujifilm XT3, crystal clear, 8K UHD, highly detailed glossy eyes, high detailed skin, skin pore

Stable Diffusion V1.5 model

Stable Diffusion XL base

Some observations:

- ➔ The SDXL model produces higher quality images.
- ➔ The SDV1.5 model does not follow prompt

2.2) Fine-Tuning using LoRA :

LoRA, or Low-Rank Adaptation, is a lightweight training technique used for fine-tuning Large Language and Stable Diffusion Models without needing full model training. Full fine-tuning of larger models (consisting of billions of parameters) is inherently expensive and time-consuming.

LoRA is a state-of-the-art fine-tuning method proposed by Microsoft researchers to adapt larger models to particular concepts. A typical complete fine-tuning involves updating the weights of the entire model in each dense layer of the neural network.

Training code of LoRA

```
accelerate launch train_text_to_image_lora_sdxl.py \
  --pretrained_model_name_or_path="stabilityai/stable-diffusion-xl-base-1.0" \
  --train_data_dir="data" --caption_column="text" \
  --resolution=1024 --random_flip \
  --train_batch_size=1 \
  --learning_rate=1e-04 --lr_scheduler="constant" --lr_warmup_steps=0 \
  --seed=42 \
  --output_dir="sd-lora-sdxl" \
  --gradient_checkpointing \
  --mixed_precision="fp16" \
  --num_train_epochs=100
```

Here we have used sdxl as base model and we will fine tune it for 100 epochs
Initially, the script will download all the required SDXL files from HuggingFace Hub and save it locally. You can find these files in the default cache folder. Subsequently, it will reuse the same cache for training.

```
|- output
|   |- checkpoint-500
|   |- checkpoint-1000
|   |- checkpoint-1500
|   |- checkpoint-2000
|- data
|- train_text_to_image_lora_sdxl.py
```

By default, each checkpoint folder consists of the following files:
optimizer.bin,pytorch_lora_weights.bin,random_states_0.pkl,scaler.pt,scheduler.bin

The pytorch_lora_weights.bin file can be used directly for inference.
one of the major advantages of LoRA is that you get excellent results by training orders of magnitude less weights than the original model size. We designed an inference process that allows loading the additional weights on top of the unmodified Stable Diffusion model weights.

First, we'll use the Hub API to automatically determine what was the base model that was used to fine-tune a LoRA model.

code

```
from huggingface_hub import model_info

# LoRA weights ~24 MB
model_path = "jaysoni/sd_xl_lora"

info = model_info(model_path)
model_base = info.cardData["base_model"]
print(model_base)   # stablediffusion_XL
```

-After we determine the base model we used to fine-tune with LoRA, we load a normal Stable Diffusion pipeline. We'll customize it with the `DPMSolverMultistepScheduler` for very fast inference:

Code

```
import torch
from diffusers import StableDiffusionPipeline, DPMSolverMultistepScheduler

pipe = StableDiffusionPipeline.from_pretrained(model_base,
torch_dtype=torch.float16)
pipe.scheduler =
DPMSolverMultistepScheduler.from_config(pipe.scheduler.config)
```

We load the LoRA weights from the Hub *on top of the regular model weights*, move the pipeline to the cuda device and run inference:

Code

```
pipe.unet.load_attn_procs(model_path)
pipe.to("cuda")
prompt = “man wearing a blue kurta, blue pant, waling down the
street, black loafer"
image = pipe( prompt, num_inference_steps=50 ).images[0]

image.save("genrated_image_lora.png")
```

2.3) ControlNet :

ControlNet was introduced in Adding Conditional Control to Text-to-Image Diffusion Models by Lvmin Zhang, Anyi Rao, and Maneesh Agrawala.

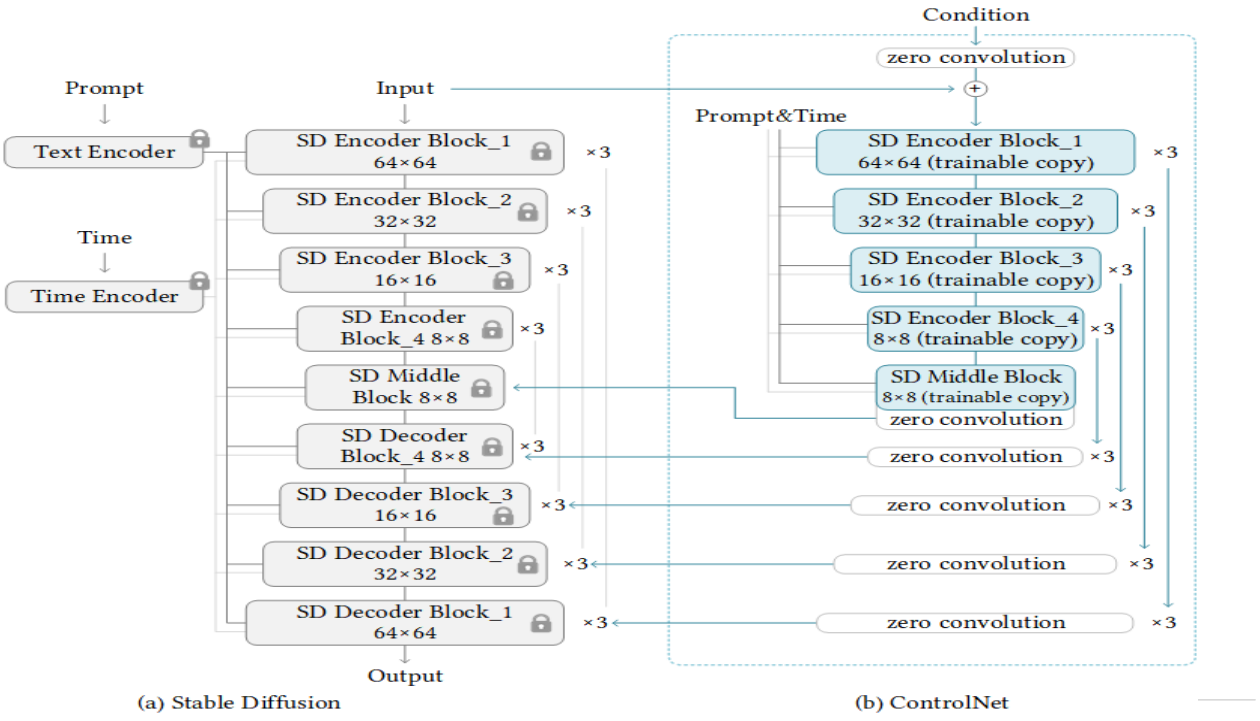
With a ControlNet model, you can provide an additional control image to condition and control Stable Diffusion generation. For example, if you provide a depth map, the ControlNet model generates an image that'll preserve the spatial information from the depth map. It is a more flexible and accurate way to control the image generation process.

Training ControlNet is comprised of the following steps:

1. Cloning the pre-trained parameters of a Diffusion model, such as Stable Diffusion's latent UNet, (referred to as "trainable copy") while also maintaining the pre-trained parameters separately ("locked copy"). It is done so that the locked parameter copy can preserve the vast knowledge learned from a large dataset, whereas the trainable Scopy is employed to learn task-specific aspects.

2. The trainable and locked copies of the parameters are connected via "zero convolution" layers (see here for more information) which are optimized as a part of the ControlNet framework. This is a training trick to preserve the semantics already learned by frozen model as the new conditions are trained.

Pictorially, training a ControlNet looks like so:



This diagram is taken from here[10]

To experiment with ControlNet, Diffusers exposes the StableDiffusionControlNetPipeline similar to the other Diffusers pipelines. Central to the StableDiffusionControlNetPipeline is the controlnet argument which lets us provide a particular trained ControlNetModel instance while keeping the pre-trained diffusion model weights the same.

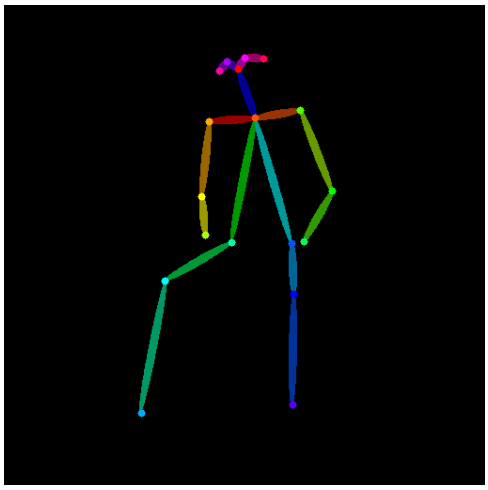
We have used Open pose ContrlNet because take a pose from one image and reuse it to generate a different image with the exact same pose.

##Code

```
from controlnet_aux import OpenposeDetector
model = OpenposeDetector.from_pretrained("lllyasviel/ControlNet")

poses = [model(img) for img in imgs]
image_grid(poses, 2, 2)
```

Input Image



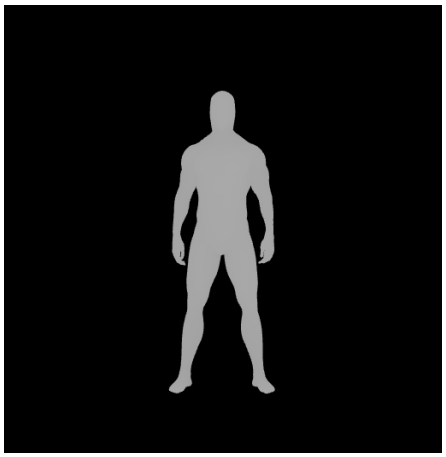
Extracted Pose

##Code (This is a controlNet pipeline based on stable diffusionv1.5)

```
controlnet = ControlNetModel.from_pretrained(
"fusing/stable-diffusion-v1-5-controlnet-openpose", torch_dtype=torch.float16
)

model_id = "runwayml/stable-diffusion-v1-5"
pipe = StableDiffusionControlNetPipeline.from_pretrained(
    model_id,
    controlnet=controlnet,
    torch_dtype=torch.float16,
)
pipe.scheduler = UniPCMultistepScheduler.from_config(pipe.scheduler.config)
pipe.enable_model_cpu_offload()
```

We can use model pose from 3D modeling website(For the visualization I have used this website : https://posemy.art/) and use that pose in Control-Net



Input pose
(https://posemy.art/)

Generated Image in same Pose

2.4) Refiner Model :

Refiner model is last step in our image generation pipeline. Refiner model is used to polish the generated image give some final touch and remove noise or any impurity from image and make it look like real image

The refiner model has been trained to denoise small noise levels of high quality data and as such is not expected to work as a pure text-to-image model; instead, it should only be used as an **image-to-image model**.

No. steps : 10

Without Refiner

No. steps : 10

With Refiner

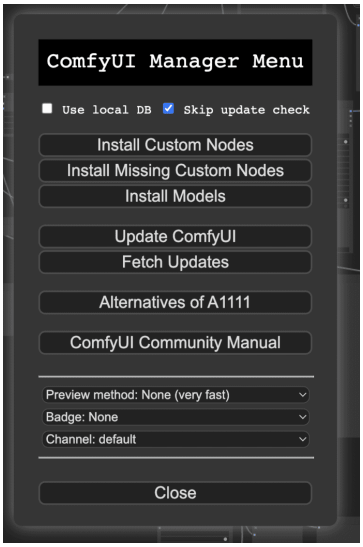
ComfyUI :

ComfyUI is a node-based GUI for Stable Diffusion. You can construct an image generation workflow by chaining different blocks (called nodes) together. Some commonly used blocks are Loading a Checkpoint Model, entering a prompt, specifying a sampler, etc. ComfyUI breaks down a workflow into rearrangeable elements so you can easily make your own.

The benefits of using ComfyUI are:

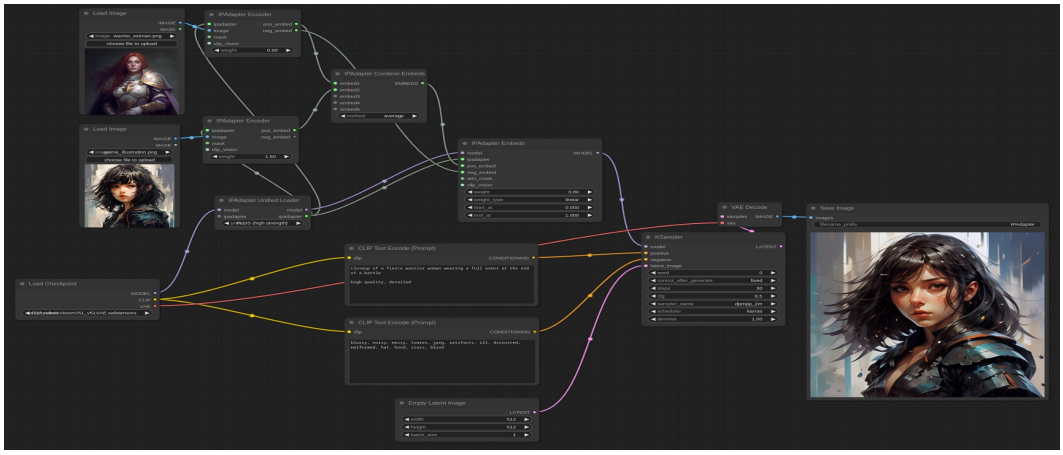
- Lightweight: it runs fast.
- Flexible: very configurable.
- Transparent: The data flow is in front of you.
- Easy to share: Each file is a reproducible workflow.
- Good for prototyping: Prototyping with a graphic interface instead of coding.

There is another main benefit of using comfy manager is that we don't have waste time in finding missing models.



Here as showcase in image that there are many option available to us where main thing to focus is that option 1 "Install Custom Nodes"and"Install Missing Custom Node" from that we can get missing model weights from our comfyUI network

There are many custom workflow available for text-to-image generation we can use that workflow. To use that workflow we only need the .json file of the workflow.



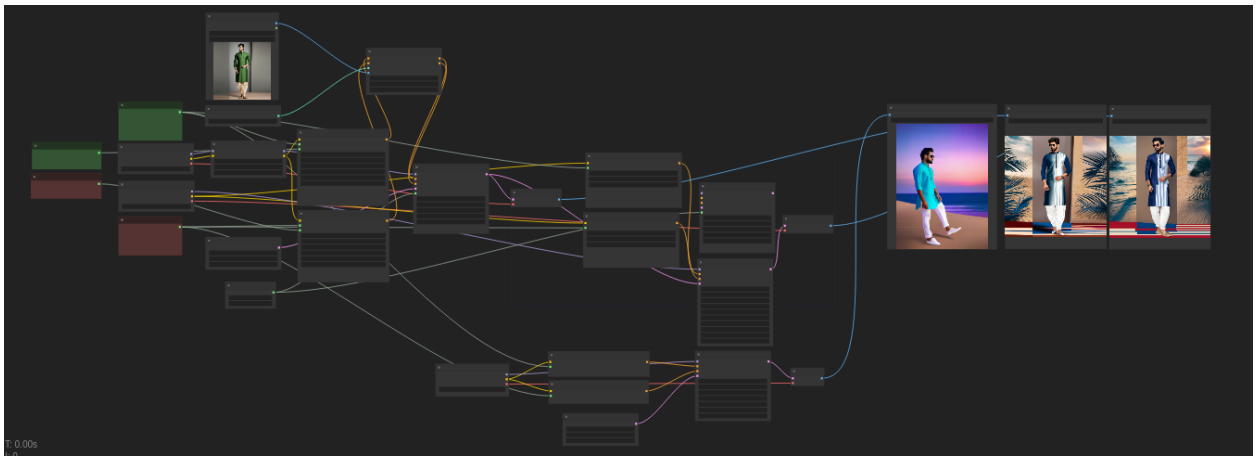
Demo workflow of sdxl in ComfyUI

Also we can share our created workflow which we have custom made to do that make a copy for workflow in json format and share that. There are two option available for sharing workflow simple workflow and api_workflow who have to use api version or use in different platform

But there are some drawbacks of using ComfyUI are:

- Inconsistent interface: Each workflow may place the nodes differently. You need to figure out what to change.
- Too much detail: Average users don't need to know how things are wired under the hood.
- Prior Knowledge : To create a custom workflow you would require some knowledge of stable diffusion and different parameter knowledge to create a good workflow
- We can't create an web app UI from it using api for that we need to get our hands dirty in code.

I have used ComfyUI to check the fine-tune models form which model workflow/pipeline Image will generate in clear and realistic I have created around 10-11 models to do fine-tuning and Uploaded to Hugging Face.
Here is the screenshot my workflow made in ComfyUI:



Since this is a large workflow with multiple model so it is not understabable from above image. But the

First image in output is from SDXL model (20 inference steps)(used to compare it with the generated image from custom LoRA)

The second Image in Output is from fine-tuned LoRA (without Refiner)

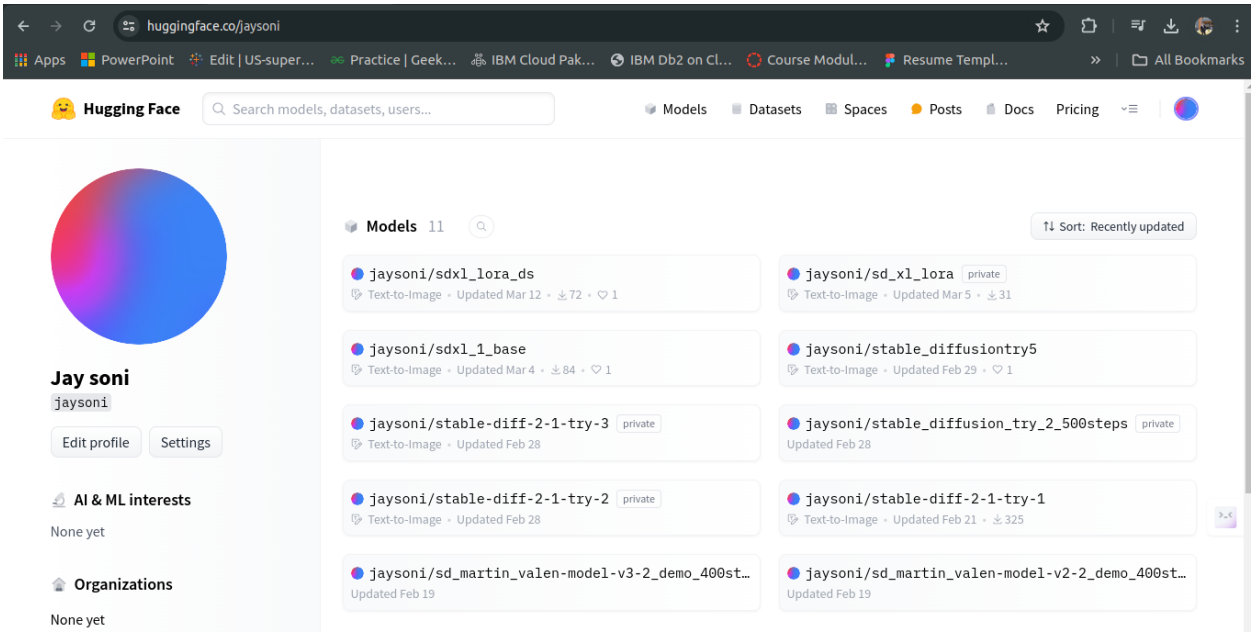
The Third Image and Last image is refined Image

Deployed models on Hugging Face :

A place where a broad community of data scientists, researchers, and ML engineers can come together and share ideas, get support and contribute to open source projects.

One of the main features of Hugging Face is the ability to create your own AI models. This model will be hosted on the platform, enabling you to add more information about it, upload all the necessary files, and keep track of versions. You can control whether your models are public or private, so you can decide when to launch them to the world or even if you'll launch them at all.

I have used Hugging Face to deploy Fine tuned models of Stable Diffusion:



As shown in the image that I have uploaded 10-11 models and tried different variation of stable diffusion model with different parameter and data from this model we haves used “jaysoni/sd_xl_lora” model because it give us satisfaction results than other models.

Created Web App using Streamlit:

Streamlit is a popular open-source Python library for building data science applications. It provides an easy-to-use interface for data analysis, visualization, and machine learning. Streamlit enables users to create interactive web applications.

##Code to run

```
!streamlit run app.py --global.logLevel=debug
```

6. Result & Discussion

In the UI there will be some option given to the use :

- Text Prompt : Here user will write what we want to Generate, we have to describe what type of image we want to generate.
- Negative Prompt : Here we have to add the things which we do not want in generated image. Like ex: Cartoon, animated, bad anatomy, naked, nude
- Inference Step : this parameter is to set the number of steps need to run for denoiseing the image (the more the better)
- Image Size : There are three image size available for generation 1024X1024, 512X512, 256X256
- Generate Button : to generate image from the above parameter
- Download Button : To download the button

Results :

(1) **Prompt** : a man wearing a pink tshirt layered with green jacket, beige pant, walking down the street, smile face, potrait iamge style
No. of Inference steps : 50

(2) **Prompt :** a man wearing a blue kurta, white pant, beige loafer, walking, wearing sunglasses, outside green environment, red sky
No. Inference step: 20

model : SDXL

model: SDXL LoRA

model: LoRA+Refiner

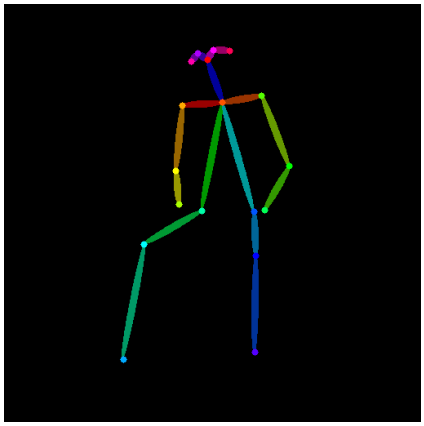
Note: As you can see in this result SDXL may generate realistic image with more number of iteration. But in less number of steps fine-tune lora and refiner is performs well.

(3) **Prompt :** a man wearing a green designer suit, flower print, beige pant, gery background, smiling face, standing pose, professional photo
No. of step : 50 , mode : LoRA + refiner

(4) **Prompt :** a man wearing a blue flower print coat and beige pant, beige background, aesthetic style, full body potrait style photography, smile on face, blue eyes, eyes facing camera, ((suede shoes))
Image :

Now I have used the above generated image in controlNet to generate new image.
Image :

Extracted Pose:



As we can see that In this image it does follow same pose but does image is not use able

Some Results of the StyleGen :

(1)

(2)

Prompt : a man wearing a blue Kurta, black loafer, professional photography, wearing a watch, light green background, camera facing
No. of Steps : 50

As we can see this is a Good results from StyleGen's results can be used in marketing or to showcase any product.

The main problem with the evaluation of Generated image is that there is not any proper method or evaluation score which says that this should be the generated image for this prompt with accuracy only thing which works is the Human Touch in Evaluation While quantitative metrics like Inception Score (IS) or Frechet Inception Distance (FID) offer insights, they don't capture the subjective experience of human perception. For text-to-image generation, and artistic exploration in particular, the "human touch" remains essential for evaluating factors for image generation tasks.

7. Conclusion

Conclusion for this project is that the styleGen represents a significant leap forward in image generation, users can explore different with artistic style and control over the creation process. By leveraging advancements in latent diffusion models, artistic style transfer, and user control mechanisms, styleGen unlocks a new door for artistic exploration and Generation.

While challenges remain in objectively evaluating the quality and accuracy of these artistic outputs, human evaluation plays a vital role. styleGen's core strengths lie in its ability to:

- Generate high-resolution images from textual descriptions.
- Infuse these images with diverse poses and images.
- Empower users to refine the artwork through control over specific visual elements and refine the image.

These functionalities position styleGen as a powerful tool for artists, designers or someone who wants to advertise their cloths on any E-comerce site they can leverage styleGen potaintial and generate professional images like photography with fine tuning prompt and different parameters:

By continuing to push the boundaries of image generation, styleGen paves the way for a future where human creativity and artificial intelligence can work together to create new and exciting forms of image generation and artistic expression.

There are some steps to be used to followed and keep in mind that while using any generative model

Understanding Limitations and Biases:

Bias Awareness: Image generation models are trained on massive datasets of images and text, which can reflect existing biases. Be mindful of potential biases in the generated content and refine your prompts accordingly.

Limited Reasoning: These models don't possess true understanding or reasoning capabilities. They excel at pattern recognition and image synthesis based on the data they're trained on. Don't expect them to grasp complex concepts or narratives perfectly.

Ethical Considerations:

Copyright and Originality: Ensure you have the rights to use any reference images you incorporate. Generated content should be used responsibly and ethically, respecting copyright and avoiding the creation of misleading or harmful material.

8. Bibliography

[1] Leon A. Gatys et al. "Image Style Transfer Using Convolutional Neural Networks" (2015).

[2] Rameen Xu et al. "Bringing Old Masters Back to Life with Controllable Latent Diffusion" (2023).

[3]"ControlNet: Interactive Control of Image Generation with Conditional Diffusion Models" (2022).

SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis, Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, Robin Rombach, arXiv:2307.01952