Fine-tuning Stable Diffusion for Style Transfer

Arkadeep Acharya
2101AI41

Project for course CS-385 Introduction to Computer Vision

Computer Science and Engineering
Indian Institute of Technology Patna

Contents

Stable Diffusion Fine-tuning - Project Report

Arkadeep Acharya

April 30, 2024

Abstract

This Project investigates the impact of fine-tuning the diffusion model for style transfer. Through experimentation, it explores how adjusting various parameters influences the fidelity and quality of style transfer results. The study aims to elucidate optimal fine-tuning hyper-parameters to enhance the performance and versatility of the diffusion model in generating stylized images using very few images.

1 Introduction

The task of fine-tuning diffusion models for style transfer holds significant relevance in the domain of computer vision and image processing. Style transfer techniques enable the seamless integration of artistic styles onto existing images, facilitating creative expression and visual enhancement. By refining diffusion models through fine-tuning, researchers aim to improve the fidelity and realism of stylized outputs, thereby expanding the utility of style transfer in various applications such as graphic design, digital art creation, photo editing, and multimedia content production.In this project we explore how we can fine-tune diffusion models using very few examples for style transfer. Github repository: https://github.com/ArkadeepAcharya/CV_Project_Style_Transfer

1.1 Codes and github repositories used for reference

- https://github.com/huggingface/transformers
- https://github.com/huggingface/diffusers

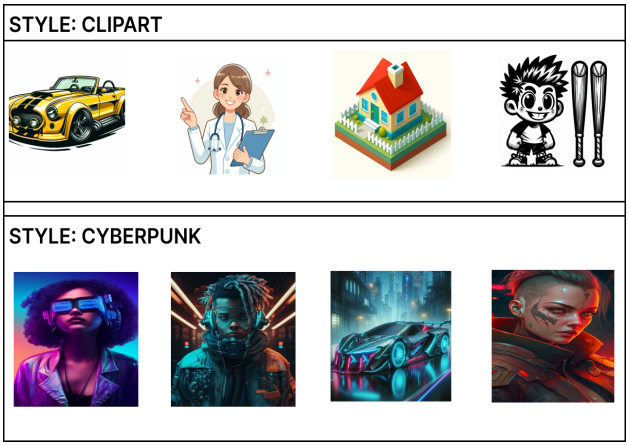

Figure 1: Style Transfer Dataset

- https://github.com/huggingface/peft

2 Style Transfer

In this section we discuss the working of stable diffusion models and how we fine-tune it using Lora adapters.

2.1 Dataset

Due to lack of good dataset for this task, I created my own dataset. My primary objective for creating the dataset was to find few but diverse images belonging to a particular style. The dataset was create by seaching for a particular style in Pinterest [1] and through images create through meticulous prompting in Bing Image Creator [2] which is powered by DALLE-3 [3] which is the state of the art model for image generation.

Examples for the created dataset for various classes has been shown in Figure 1.

[1] https://in.pinterest.com/
[2] https://www.bing.com/images/create
[3] https://openai.com/dall-e-3

2.2 Fine Tuning of Stable Diffusion

2.2.1 The Stable Diffusion Model

At its essence, Stable Diffusion constitutes a sophisticated deep learning model harnessing diffusion processes to generate high-fidelity artwork from input descriptions. Functionally, it operates by synthesizing realistic images corresponding to provided textual prompts.

The diffusion process redistributes pixel values based on local information – reducing noise by diffusing pixel values in smooth regions while preserving sharp transitions and edges.

The entire process involves several stages:

- **Interpretation**: Users input a description in natural language, which Stable Diffusion interprets and analyzes through artificial intelligence to extract relevant information.
- **Diffusion Model**: Stable Diffusion employs a diffusion model trained to eliminate Gaussian noise from blurry images, refining them iteratively until a sharp and clear result is achieved.
- **Continuous Learning**: With each interaction, Stable Diffusion learns and improves its outputs over time, progressively generating more precise and lifelike images.

In my experiment I have used the *runwayml/stable-diffusion-v1-5*[4]. This model uses a CLIP [3] based text-encoder which has been specifically train for this task for prompt interpretation, and a conditional variational auto encoder that has been conditioned on the prompt and a 2D conditional Unet [5] to iteratively remove noise from a randomly initialized noisy image to generate image which are aligned to the user prompt.

2.2.2 FineTuning using PEFT and LoRA

Fine-tuning large pre-trained models is often prohibitively costly due to their scale. Parameter-Efficient Fine-Tuning (PEFT) methods enable efficient adaptation of large pre-trained models to various downstream applications by only fine-tuning a small number of (extra) model parameters instead of all

[4] https://huggingface.co/runwayml/stable-diffusion-v1-5
[5] https://huggingface.co/docs/diffusers/en/api/models/unet2d-cond

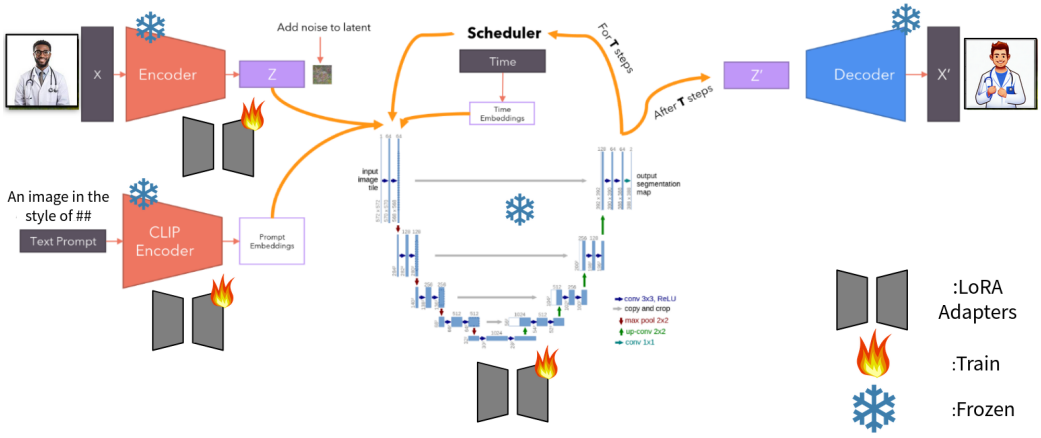

Figure 2: Architecture for Fine-tuning stable diffusion for Style transfer.

the model's parameters. This significantly decreases the computational and storage costs. Recent state-of-the-art PEFT techniques achieve performance comparable to fully fine-tuned models.

LoRA [2], which stands for "Low-Rank Adaptation", distinguishes itself by training and storing the additional weight changes in a matrix while freezing all the pre-trained model weights. LoRA is not called an "adapter" because it does not add adapters. Instead, it is referred to as "adaptation" to describe the process of fine-tuning the domain data and tasks. The key innovation of LoRA lies in decomposing the weight change matrix Δ W into two low-rank matrices, A and B. Instead of directly training the parameters in Δ W, LoRA focuses on training the parameters in A and B matrixes. The fine-tuning is shown in the figure 3.

For fine-tuning my go to approach was using Lora [1] adapters on the attentions in all of the 3 components, namely the text encoder, conditional VAE and conditional 2D U-Net for an efficient fine-tuning. The complete architectural diagram of model has been show in Fig 2. Lora adapter comes as an obvious choice over traditional fine-tuning due to the following reasons:

- As discussed above Peft using Lora helps use fine-tune pre-trained model for a particular task in a low resource environment. Owing to this method I could easily fine-tune the stable diffusion model in less than 7GB of GPU memory.

Hyperparameter	Value
LoRA Rank	8
LoRA Alpha	16
LoRA Dropout	0.05
Batch Size	4
Learning Rate	1e-4
Epochs	100
Target Modules (C-Unet)	to_q, to_v, query, value
Target Modules (Text Encoder)	q_proi, v_proj
Target Modules (C-VAE)	to_q, to_v,to_k

Table 1: Table showing all the hyper parameters for fine tuning ViT.

- It has been well established in literature that Peft helps us counter catastrophic forgetfulness. This becomes very crucial for this task as we do not have a well structured large enough dataset for this task and Lora adapter helps us in preserving the original weight and thus the original creative capabilities of the pre-trained model.

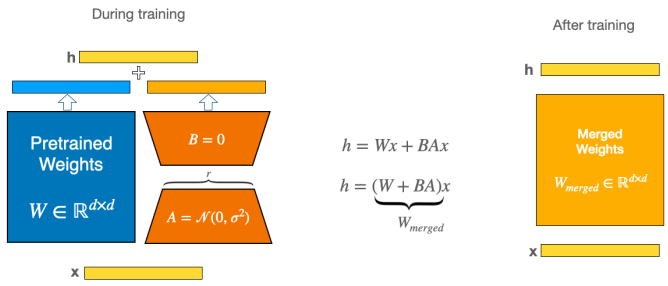

Figure 3: PEFT Fine-tuning using LORA (taken from Huggingface[6])

The hyperparameters used while fine-tuning is listed in the Table 1

3 Results

In this section I show the outputs of the model in 4 fine-tuning scenarios:

- All 3 major modules, i.e. text-encoder, Conditional VAE and Conditional U-Net has been fine-tuned.

Figure 4: Initial images for Style Transfer.

- Without Fine-tuning the text-encoder.
- Without Fine-tuning the VAE
- Without Fine-tuning the U-Net

The initial images that have been used for the experimentation of style transfer has been show in Figure 4

In the following Figures are provide some sample output for both Image generation and Style transfer in each of the following cases for 2 styles,i.e. **Clipart** and **Cyberpunk**.

The prompts for the "Image Generation"

- A photo of a doctor in the style of **.
- A photo of a car in the style of **.
- A photo of a boy in the style of **.

The prompt for the “Style Transfer”

- A photo in the style of **.

NOTE: I used “**” and not the words “Clipart” or “Cyberpunk” directly as I wanted to ensure that the model did not have any previous knowledge of the style from its pre-training and I wanted to ensure that I do-not give any hint about the style directly in the prompt. Any other special character or even my name could have been used instead of “**” during training and it would have yielded similar results.

3.1 Clipart

Figure 5 shows the output of the model for each of the 4 cases of fine-tuning for the style 'Clipart'.

3.2 Cyberpunk

Figure 6 shows the output of the model for each of the 4 cases of fine-tuning for the style 'Cyberpunk'

4 Conclusion

In conclusion, my project on style transfer utilizing stable diffusion underscores the nuanced impact of fine-tuning various components within the model architecture. Specifically, we observed that while fine-tuning the text encoder significantly influences image generation tasks, its effect is relatively diminished in the context of style transfer between images. Conversely, fine-tuning the UNet emerges as the paramount factor for ensuring consistency in output quality across both image generation and style transfer tasks. Additionally, we highlight the indispensable role of fine-tuning the VAE in enhancing output clarity and stability. These findings illuminate critical avenues for optimizing stable diffusion-based style transfer systems, paving the way for more robust and versatile artistic expression in digital media.

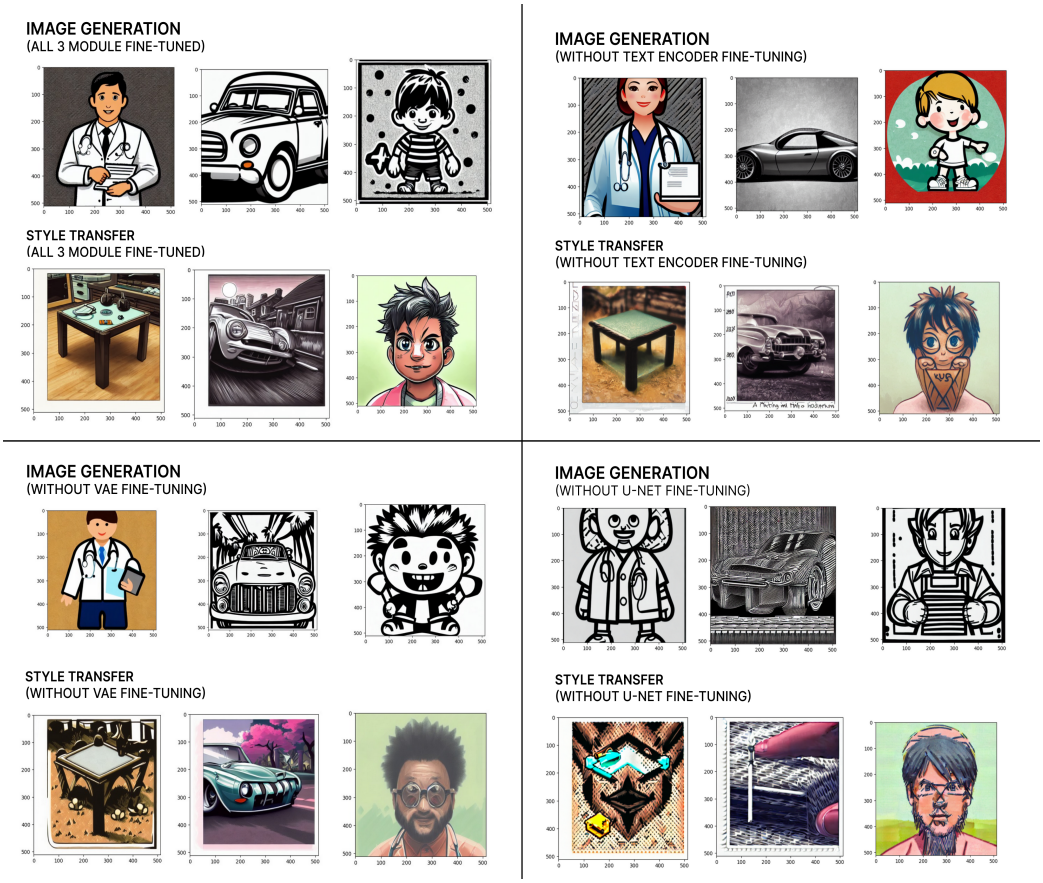

Figure 5: Outputs generated by the model for the style **Clipart**.

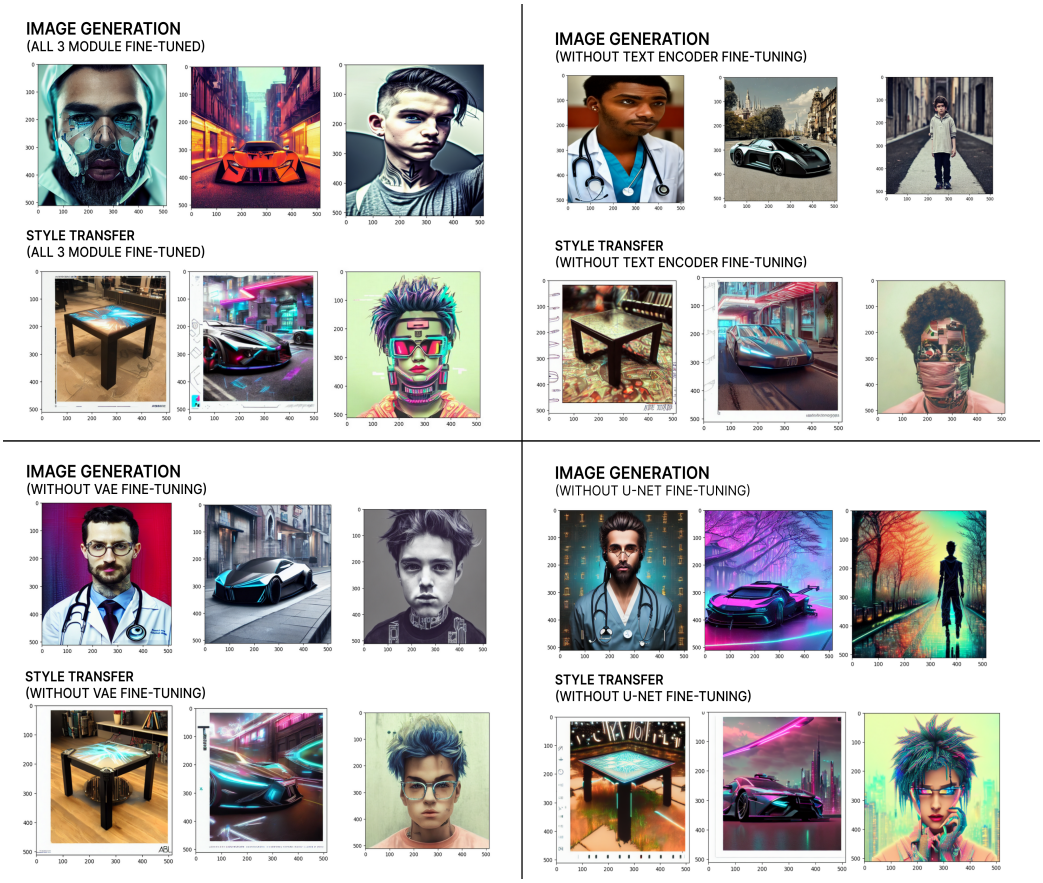

Figure 6: Outputs generated by the model for the style **Cyberpunk**

References

[1] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL].

[2] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *CoRR* abs/2106.09685 (2021). arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[3] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].