Journal of Digital Art Engineering & Multimedia Vol.10, No.4, December (2023), pp.567-576
ISSN 2508-9099(p) 2672-1198(e) *http://dx.doi.org/10.29056/jdaem.2023.12.12*

Improved DreamBooth: SNR 조정 및 노이즈 스케줄러 최적화를 통한 개선된 파인튜닝

Improved DreamBooth: Enhanced Fine-Tuning through SNR Adjustment and Optimization of Noise Scheduler

조원지[1], 박성욱[2], 정세훈[3], 심춘보[4]*

Won-ji Jo[1], Sung-Wook Park[2], Se-hoon Jung[3], Chun-bo Sim[4]*

요 약

Text-to-Image 생성은 자연어 설명을 기반으로 해당 설명과 근사한 이미지를 합성한다. 그러나 대규모 Text-to-Image 모델은 특정 참조 세트 내 대상(Subject)의 외형을 모방하고, 다른 맥락(Context)에서 새로운 표현을 합성하는 능력이 부족하다. 이러한 한계를 극복하고자 DreamBooth는 Text-to-Image Diffuison 모델을 사용자 맞춤형으로 개인화(Personalization)하기 위한 파인튜닝 방법을 제안한다. DreamBooth를 통해 개인화된 Stable Diffusion 또한, 간혹 얼굴 합성 능력이 부족한 문제가 발생한다. 이에 따라 본 논문에서는 신호 대 잡음비(SNR; Signal to Noise Ratio) 조정 알고리즘을 도입하고, Beta Schedule 방식 변경 방법을 제안한다. 나아가 선행 연구와 정확한 성능 비교를 위해, 동일한 Text Prompt를 사용하여 다양한 실험을 진행했다. 실험 결과, SNR 조정 알고리즘과 Beta Schedule 방식으로 Sigmoid Schedule을 사용했을 때 가장 우수한 성능을 보였으며, Stable Diffusion의 U-Net 구조만 파인튜닝해도 효과적인 개인화가 가능함을 입증했다.

핵심어 : 텍스트 대 이미지 생성, 스테이블 디퓨전, 파인튜닝, 드림부스, 노이즈 스케줄링

Abstract

Text-to-image generation produces images based on natural language descriptions. However, large-scale text-to-image models often struggle to mimic the appearance of objects within a specific reference set and generate new representations in various contexts. To address these limitations,

1 Department of Artificial Intelligence Engineering, Sunchon National University, Sunchon, Korea [Undergraduate Student]
e-mail: wonji1283@naver.com
2 Interdisciplinary Program in IT-Bio Convergence System, Sunchon National University, Sunchon, Korea [Graduate Student]
e-mail: 411050@scnu.ac.kr
3 Department of Computer Engineering, Sunchon National University, Sunchon, Korea [Professor]
e-mail: shjung@scnu.ac.kr
4 Interdisciplinary Program in IT-Bio Convergence System, Sunchon National University, Sunchon, Korea [Professor]
e-mail: cbsim@scnu.ac.kr (Corresponding author)

* 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2022년도 문화기술 연구개발 사업으로 수행되었음(과제명 : 확장현실 융합 시스템 솔루션 연구개발 기반 문화기술 전문인력 양성, 과제번호 : R2022020014, 기여율 : 100%)

Received(November 4, 2023), Review Result(1st: November 22, 2023), Accepted(December 8, 2023), Published(December 31, 2023)

DreamBooth proposes a fine-tuning method for personalizing text-to-image diffusion models. Personalized stable diffusion through DreamBooth sometimes encounters issues with insufficient facial synthesis capabilities. Therefore, in this paper, we introduce a signal-to-noise ratio (SNR) adjustment algorithm and suggest changing the Beta Schedule method to the Sigmoid Schedule. Additionally, various experiments were conducted using the same text prompts for an accurate performance comparison with previous studies. The experimental results demonstrated the best performance when the SNR adjustment algorithm and Sigmoid Schedule were used together. Furthermore, it was proven that effective personalization is achievable by fine-tuning only the U-Net structure of Stable Diffusion.

Keyword : Text-to-Image Generation, Stable Diffusion, Fine-Tuning, Dreambooth, Noise Scheduling

1. 서론

Generative AI는 훈련 데이터의 패턴과 구조를 학습하여 새롭고 독창적인 콘텐츠를 생성하는 AI 기술이다. ChatGPT나 Stable Diffusion과 같은 대규모 생성 모델은 질의에 대한 응답이나 예술적인 이미지 합성을 포함한 다양한 작업을 수행할 수 있다. 이러한 기술을 통해 생성 모델이 산업과 사회에 미치는 영향은 지속해서 증가하고 있다 [1].

Text-to-Image 생성은 Generative AI의 한 예로, 자연어 설명을 기반으로 해당 설명과 근사한 이미지를 합성한다. 그러나 대규모 Text-to-Image 생성 모델들은 특정 참조(Reference) 세트 내 대상의 외형을 모방하고, 다른 맥락에서 새로운 표현을 합성하는 능력은 부족하다 [2]. DreamBooth는 이러한 한계를 극복하고자 Text-to-Image Diffusion 모델을 사용자 맞춤형으로 개인화하기 위한 파인튜닝 방법을 제안한다.

Stable Diffusion은 LMU(Ludwig Maximilian University) 뮌헨의 CompVis(Computer Vision & Learning Group)에서 개발한 오픈소스 LDM(Latent Diffusion Model)이다. 그러나 Stable Diffusion에 DreamBooth 방법을 적용했을 때, 간혹 얼굴이 잘 합성되지 않는 문제가 발생한다. 이는 DreamBooth에서 베이스라인 모델로 사용한 Imagen에는 SR(Super-Resolution) 컴포넌트의 파인튜닝 과정이 존재하지만, Stable Diffusion에는 SR 컴포넌트 파인튜닝 과정이 생략되며, U-Net 구조만을 파인튜닝하기 때문으로 사료된다.

본 논문에서는 Stable Diffusion의 U-Net 구조만을 파인튜닝하여 고품질의 다양한 이미지 합성이 가능한지 검토한다. 특히 얼굴 합성 능력 저하 문제를 해결하기 위해 Noise Scheduler의 Forward Process에 새로운 알고리즘을 도입하고, 이에 따른 Beta Schedule 방식을 최적화하는 방법을 제안한다. 실험 결과, SNR 조정 알고리즘과 Sigmoid Scheduler을 함께 사용했을 때 가장 성능이 우수했으며, 선행 연구에서 발견되었던 이미지 뭉개짐과 어두운 색감만을 표현하는 문제를 개선했다.

본 논문의 주요 기여는 다음과 같다:

선행 연구에서 나타난 문제점을 파악하고, Stable Diffusion을 효과적으로 개인화하기 위한 혁신적인 접근 방식을 제안한다.

제안한 방법은 다른 구성요소의 추가적인 파인튜닝 없이도 고품질 이미지 합성이 가능하다.

2. 관련 연구

2.1 DreamBooth 파인튜닝 방법

대상 이미지 3~5장을 이용하여 사전 훈련된 Text-to-Image Diffusion 모델을 두 단계로 파인튜닝한다. 첫 번째 단계에서는 Text-to-Image 모델을 저해상도 입력 이미지와 함께 파인튜닝한다. 각 이미지에는 고유 식별자(Unique Identifier)와 대상이 속한 클래스명이 담긴 Text Prompt가 포함된다. 동시에 클래스별 사전 보존 손실(Class Specific Prior Preservation Loss)을 적용하여 모델이 해당 클래스에 대한 의미론적(Semantic) 사전 지식을 활용하고, 클래스명을 사용하여 다양한 인스턴스(Instance)를 합성할 수 있도록 유도한다. 두 번째 단계에서는 입력 이미지 세트에서 추출한 저해상도 및 고해상도 이미지 쌍을 사용하여 SR 컴포넌트를 파인튜닝한다. 이로써 대상의 작은 세부 정보에 대한 높은 합성 정확도를 유지한다 [2].

2.2 Latent Diffusion Model

Denoising Autoencoder 기반의 DM(Diffusion Model)은 픽셀 공간에서 작동하기 때문에 높은 컴퓨팅 자원을 요구한다. 이 문제를 해결하기 위해 LDM은 Autoencoder에서 추출한 압축된 Latent Feature Map을 활용한다. 이는 고차원 공간이 아닌, 저차원 공간에서 샘플링을 수행하여 보다 계산 효율적인 DM을 획득할 수 있다. 또한, U-Net에서 상속받는 DM의 귀납적 편향(Inductive Bias)을 활용하여 이전 접근법에서 요구되는 품질 감소 압축 수준을 완화한다. U-Net 백본(Backbone)을 강화하기 위해, 크로스 어텐션 메커니즘(Cross Attention Mechanism)을 도입하여 DM보다 유연한 조건부 이미지 합성기 전환이 가능하며, 우수한 성능을 보인다 [3].

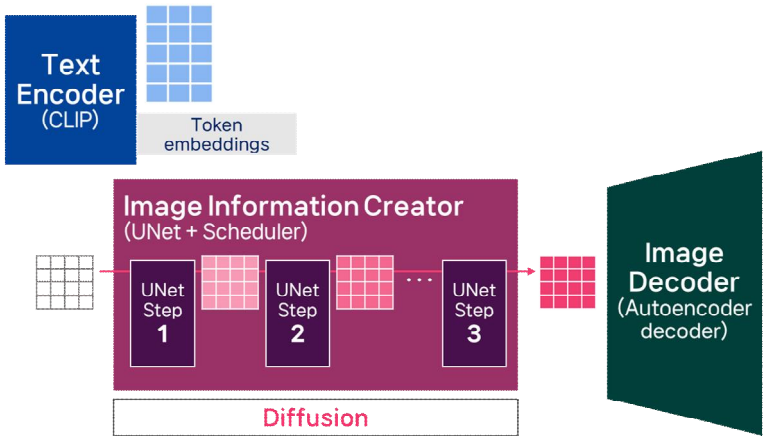
2.3 Diffusion Model의 결함과 해결 방안

Diffusion Noise Schedule 및 일부 Diffusion 샘플러의 결함으로 인해, 종종 모델의 훈련과 추론(Inference) 간 불일치가 발생한다. 특히 Stable Diffusion에서는 모델이 중간 밝기의 이미지만 합성하고, 매우 밝거나 어두운 이미지를 잘 합성하지 못하는 문제가 발생한다. 해당 문제를 해결하기 위해 Noise Schedule을 재조정하여 종단 SNR을 0으로 강제하고, 모델을 ν 예측과 함께 훈련시킨다. 또한 샘플러를 최종 타임 스텝에서 시작하도록 고정하고, 과대 노출(Over Exposure) 방지를 위해 Classifier Free Guidance를 조정한다. 이와 같은 방법은 모델이 원본 데이터 분포에 더 충실한 샘플을 생성할 수 있도록 한다 [4].

3. 제안하는 방법

3.1 제안하는 방법 개요

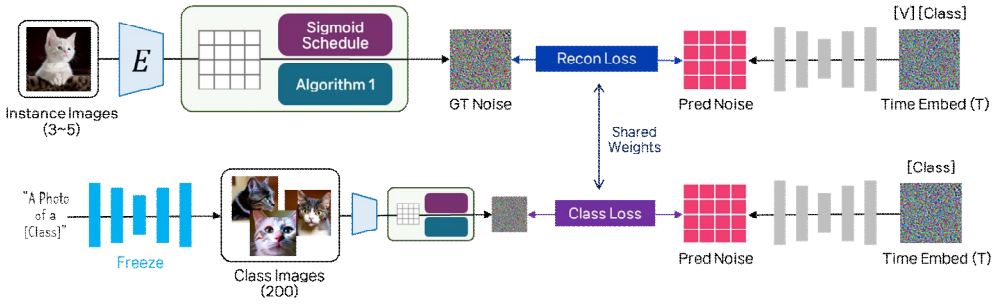
Stable Diffusion은 [그림 1]과 같이 세 가지 구성요소를 가진다. CLIP(Contrastive Language-Image Pre-training) [5]은 LDM의 Text-to-Image 모델로의 전환을 위한 Text Encoder다. Text Encoder는 문장을 Tokenizer를 통해 처리하여 Text Embedding Vector를 생성한다. U-Net에서는 Text Embedding Vector를 기반으로, Conditioning된 상태에서 Random Latent Vector를 여러 번 반복하여 Denoising한다. 이 과정에서 Noise를 어떻게 처리할지 결정하는 것이 Noise Scheduler의 주된 역할이다. 대표적인 Noise Scheduler로는 DDPM(Denoising Diffusion Probabilistic Model) [6]과 DDIM(Denoising Diffusion Implicit Model) [7]이 있다. 이후에는 생성된 저해상도의 Latent Feature Map을 사전 학습된 Autoencoder의 Decoder를 사용하여 고해상도 이미지로 변환한다.



[그림 1] Stable Diffusion의 구성요소

[Fig. 1] The Components of Stable Diffusion

Noise Scheduling 전략은 Denoising DM 성능에 중요한 역할을 하며, 최적의 Noise Scheduling 전략은 특정 작업이나 이미지 크기에 따라 다를 수 있다 [8]. 따라서 본 논문에서는 DreamBooth 방법을 이용한 Stable Diffusion의 파인튜닝 과정에서 사용되는 Noise Scheduler의 순방향 프로세스(Forward Process)에 SNR 조정 알고리즘을 추가하고, Beta Schedule 방식을 변경하는 방법을 제안한다.



[그림 2] 제안하는 방법

[Fig. 2] Proposed Method

제안하는 방법은 [그림 2]와 같다. Stable Diffusion 모델에 인스턴스 이미지 3~5장을 입력하면 Autoencoder의 인코더를 통과하여 순방향 프로세스를 거친다. 이때 Sigmoid Schedule과 알고리즘 1이 적용된다. 이후 역방향 프로세스에서는 랜덤한 Noise와 함께 해당 타임 스텝에 대한 정보와 고유 식별자 '[V]', 인스턴스 이미지의 클래스 이름 '[Class]'가 포함된 Text Embedding Vector가 주어진다. U-Net은 해당 정보를 활용하여 예측하며, 예측한 Noise와 실제 순방향 프로세스에서의 Noise를 비교하여 Reconstruction Loss를 계산한다. 클래스별 사전 보존 손실 계산에서는 고유 식별자를 제외하고 '[Class]'만을 포함한 Text Prompt를 작성하여 200장의 이미지를 합성한다. 이를 통해 기존 Stable Diffusion이 클래스명에 대해 합성하던 다양성을 보존하게 된다. Reconstruction Loss와 클래스별 사전 보존 손실은 서로 공유된다.

3.2 SNR 조정 알고리즘

알고리즘 1이 동작하는 Stable Diffusion의 순방향 프로세스는 DM [7][9]과 동일하다. 순방향 프로세스에서는 Beta Schedule인 $\beta_1, \beta_2, \cdots \beta_T$에 따라 이미지에 Noise를 추가하며, 식 1을 따른다. 여기서 α_t는 $1-\beta_t$로 정의되며, $1-\beta_t$은 첫 번째 타임 스텝부터 t번째 타임 스텝까지의 모든 곱으로 정의된다. 이에 따라 SNR은 식 2를 통해 계산한다.

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon,\ \epsilon \sim N(0,I) \tag{1}$$

$$SNR(t) = \frac{\overline{\alpha}_t}{1-\overline{\alpha}_t} \tag{2}$$

Stable Diffusion을 포함한 Diffusion 모델들은 통상 Diffusion Noise Schedule에 결함이 존재한다. 이는 최종 타임 스텝 T에서 SNR이 0에 도달하지 못한다는 것을 의미하며, 알고리즘 1을 통해 0으로 강제함으로써 해결한다 [4].

Algorithm 1 Pseudocode of Zero Terminal SNR in a TF-like style

```
def enforce_zero_terminal_snr(self, betas):
    # Convert betas to alphas_bar_sqrt
    alphas = 1.0 - betas
    alphas_bar = tf.math.cumprod(alphas)
    alphas_bar_sqrt = tf.sqrt(alphas_bar)

    # Store old values.
    alphas_bar_sqrt_0 = alphas_bar_sqrt[0]
    alphas_bar_sqrt_T = alphas_bar_sqrt[-1]
    # Shift so last timestep is zero.
    alphas_bar_sqrt -= alphas_bar_sqrt_T
    # Scale so first timestep is back to old value.
    alphas_bar_sqrt *= alphas_bar_sqrt_0 / (alphas_bar_sqrt_0 - alphas_bar_sqrt_T)

    # Convert alphas_bar_sqrt to betas
    alphas_bar = alphas_bar_sqrt ** 2
    alphas = alphas_bar[1:] / alphas_bar[:-1]
    alphas = tf.concat([alphas_bar[0:1], alphas], axis=0)
    betas = 1 - alphas
    return betas
```

3.3 Sigmoid Schedule

Beta Schedule은 기본적으로 0.0001에서 0.02로 Beta를 점진적으로 증가시키는 Linear Schedule에 해당한다. 최근에는 Linear Schedule이 순수한 Noise로의 전환이 너무 급격하게 진행된다는 문제가 있어, 이를 완화하기 위해 Cosine Schedule을 사용기도 한다 [10].

본 논문에서는 SNR 조정 알고리즘과의 조합에서 Beta Schedule 방식으로 Linear Schedule 대신 Sigmoid Schedule을 사용했을 때 성능이 가장 우수한 것으로 나타났다. 따라서, Beta Schedule 방식을 Sigmoid Schedule로 변경한다.

4. 실험 및 결과

4.1 실험 데이터 세트 구성 및 환경설정

입력 이미지로 사용한 인스턴스 이미지는 [그림 3]과 같고, 5장의 이미지는 생성 모델로 구축한

인종 얼굴 데이터 세트 [11]에서 선정한 가상 인물이다.

[그림 3] 인스턴스 이미지

[Fig. 3] Instance Images

파인튜닝 하기 전 Stable Diffusion을 활용하여 'A photo of person without mustache, handsome, ultra realistic, 4k, 8k'와 같이 Text Prompt를 작성해 클래스 이미지를 준비했다. 선행 연구에 따르면, 200번의 에포크는 좋은 결과를 얻기 위해 충분했다 [2]. 따라서 클래스 이미지는 총 200장을 준비했다. Text Prompt는 단순히 'A photo of person'으로 사용했을 때보다, 'mustache'나 'handsome'과 같은 단어를 추가했을 때 성별을 더 정확하게 반영했다. 이는 해당 단어들을 사용함으로써 여성이 아닌 남성의 이미지를 더 잘 합성하도록 유도한 것이며, 샘플은 [그림 4]와 같다.

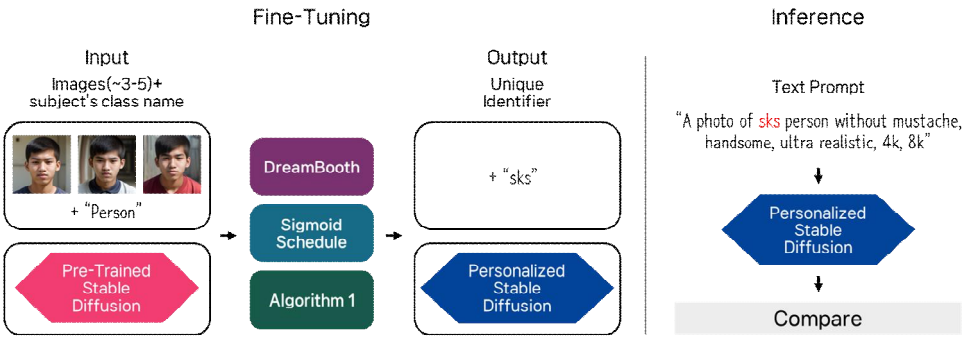
[그림 4] 합성한 클래스 이미지 샘플

[Fig. 4] Samples of Synthesized Class Images

고유 식별자는 인스턴스 이미지의 특정 대상을 다른 대상들과 구별하는 데 사용된다. 기존의 영어 단어를 고유 식별자로 사용하는 것도 가능하지만, 이는 Text-to-Image 모델이 사전에 학습한 데이터 세트로 인해 특정 대상이 지닌 의미보다 기존 의미에 대해 더 강력하게 작동할 수 있다 [2]. 본 실험에서는 고유 식별자로 어떤 의미도 없는 'sks' 단어를 선택했다.

4.2 실험 방법

[그림 5]는 제안하는 방법과 기존 방법의 성능을 비교하기 위한 실험과정의 개요다. 기존 방법과 알고리즘 1을 추가한 경우, Sigmoid Schedule을 사용한 경우, 그리고 Sigmoid Schedule과 알고리즘 1을 함께 사용한 경우를 비교했다. 정확한 비교를 위해 추론 과정에서 Text Prompt는 'A photo of sks person without mustache, handsome, ultra realistic, 4k, 8k'로 통일했다.



[그림 5] 성능 비교를 위해 진행한 파인튜닝 및 추론 방법의 개요

[Fig. 5] An overview of the fine-tuning and inference processes

파인튜닝 과정 중 사용되는 하이퍼파라미터로 최종 타임 스텝은 1000, Learning Rate는 5e-6로 설정했고, 실험 환경은 [표 1]과 같다.

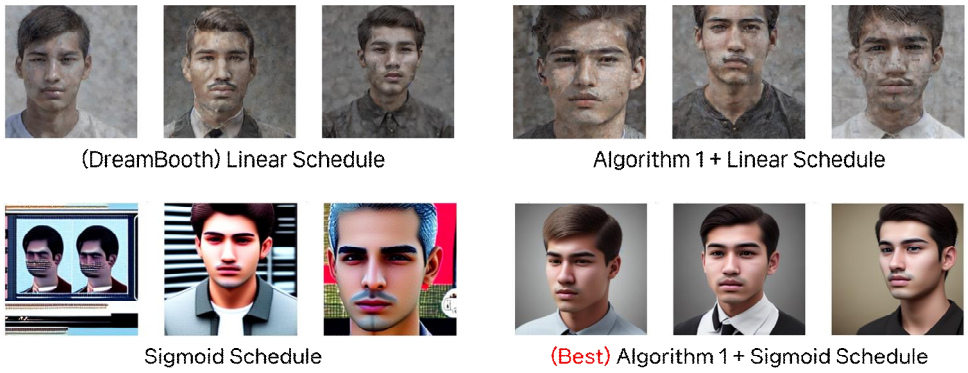
[표 1] 실험 환경

[Table 1] Experimental Environment

Hardware	CPU	Intel Core i7-12700KF
	GPU	NVIDIA GeForce RTX 3090 Ti 24GB
	RAM	Samsung DDR4 32GB
	Operating System	Ubuntu Linux 20.04.6 LTS
Software	CUDA	11.8.0
	cuDNN	8.6.0
	Programming Language	Python 3.10.12
	Deep Learning Library	Tensorflow 2.12.0

4.3 실험 결과

[그림 6]은 실험 결과로, 네 가지 경우 모두 Stable Diffusion의 U-Net만을 파인튜닝한 것이다. 기존 방법에 알고리즘 1을 추가했을 때는 성능이 크게 향상되지 않았다. 기존 Linear Schedule 대신 Sigmoid Schedule을 도입한 경우, 다양한 색조의 합성이 이뤄졌지만 충실도(Fidelity)는 낮게 관찰됐다. 알고리즘 1과 Sigmoid Schedule을 조합한 결과, 가장 뚜렷한 성능 향상을 보여주었으며, 이미지 뭉개짐 및 어두운 색조만을 표현하는 문제가 개선됐다.



[그림 6] 다양한 조합으로 실험한 얼굴 이미지 합성 결과

[Fig. 6] Results of facial image synthesis with various combinations in the experiments

앞선 실험에서는 'handsome'과 같이 주관적 평가가 개입될 가능성이 농후한 단어를 사용했기 때문에, 특정 인물의 고유한 특징을 보존하는 데 제한이 있을 수 있다. 따라서 본 논문에서는 비교적 주관성이 낮은 단어를 사용하여 인스턴스 이미지의 특징을 얼마나 효과적으로 보존하는지 육안으로 식별해 비교했다. [그림 7]을 통해 주관성이 적은 단어를 사용했을 때, 특정 인물의 고유한 특징을 더 효과적으로 보존할 수 있음을 확인했다.

[그림 7] 그 외 다양한 텍스트 프롬프트로 실험한 얼굴 이미지 합성 결과

[Fig. 7] Facial image synthesis results from experiments with various other text prompts

5. 결론

본 논문에서는 DreamBooth 방법을 이용한 Stable Diffusion의 U-Net 파인튜닝을 효과적으로 수행하기 위해 SNR 조정 알고리즘과 Sigmoid Schedule을 결합하는 방법을 제안한다. 실험 결과, SNR

조정 알고리즘과 Sigmoid Schedule을 각각 도입했을 때 선행 연구에서 발견되었던 문제점이 크게 개선되지 않았지만, 두 가지를 함께 사용했을 때는 성능이 향상된 것으로 확인됐다. 이로써 Stable Diffusion의 Text Encoder나 다른 구성요소의 추가적인 파인튜닝 없이도 Text-to-Image 모델의 효율적인 개인화가 가능하며, 그에 따른 컴퓨팅 자원도 절약될 것으로 기대된다. 또한, 다양한 사용자들이 본인들 목적에 맞는 Text-to-Image 작업을 수행할 수 있을 것으로 사료된다. 향후 다양한 생성 모델 파인튜닝 방법들의 상하 관계를 명확하게 밝힐 수 있는 연구가 필요하다 [12].

References

[1] R. Gozalo-Brizuela, E. C. Garrido-Merchan, “ChatGPT is not all you need. A State of the Art Review of large Generative AI models”, unpublished.

[2] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, K. Aberman, “Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation”, IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 17-24, 2023, Vancouver, BC, Canada, pp. 22500-22510, doi: 10.1109/CVPR52729.2023.02155.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, “High-resolution image synthesis with latent diffusion models”, IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-24, 2022, New Orleans, LA, USA, pp. 10674-10685, doi: 10.1109/CVPR52688.2022.01042.

[4] S. Lin, B. Liu, J. Li, X. Yang, “Common Diffusion Noise Schedules and Sample Steps are Flawed”, unpublished.

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, “Learning Transferable Visual Models From Natural Language Supervision”, International Conference on machine learning, July 18-24, 2021, Online, pp. 8748-8763.

[6] J. Ho, A. Jain, P. Abbeel, “Denoising Diffusion Probabilistic Models”, Advances in Neural Information Processing Systems, vol. 33, May 2020, pp. 6840-6851.

[7] J. Song, C. Meng, S. Ermon, “Denoising Diffusion Implicit Models”, International Conference on Learning Representations, May 3-7, 2021, Online.

[8] C. Ting, “On the Importance of Noise Scheduling for Diffusion Models”, unpublished.

[9] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics”, International Conference on Machine Learning, Lille, France, July 7-9, 2015, pp. 2256-2265.

[10] A. Q. Nichol, P. Dhariwal, “Improved Denoising Diffusion Probabilistic Models”, International Conference on Machine Learning, July 18-24, 2021, Online, pp. 8162-8171.

[11] J. Lee, “Google Drive”, Google, https://drive.google.com/drive/folders/1DeZWQhnxMkoiqlM8ylWB9_N4J1AMzCYF, (accessed September 7, 2023).

[12] S. W. Park, J. Y. Kim, J. Park, S. H. Jung, C. B. Sim, “How to train your pre-trained GAN models”, Applied Intelligence, vol. 53, no. 22, August 2023, pp. 27001-27026, doi: 10.1007/s10489-023-04807-x.