Jurnal Teknik Informatika Vol. 17 No. 1, April 2024 (21-29)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i1.37942

JURNAL TEKNIK INFORMATIKA

*Homepage* : http://journal.uinjkt.ac.id/index.php/ti

# The Comparison of the Effectiveness and Efficiency of Fine-Tuning Models on Stable Diffusion in Creating Concept Art

**Bilal Abdul Qowy[1], Ahmad Nur Ihsan Purwanto[2*] and Sri Hartati[3]**

[1,2,3]Computer Science, UAG University
[1,2,3] Tower 165 Jl. TB. Simatupang Lot 1 Cilandak, South Jakarta, Indonesia

**Article:**
Accepted: February 02, 2024
Revised: November 23, 2023
Issued: April 30, 2024

***Correspondence Address:**
ahmadnur.ihsan@esqbs.ac.id

## ABSTRACT

This research aims to overcome the limitations of the Stable Diffusion model in creating conceptual works of art, focusing on problem identification, research objectives, methodology and research results. Even though Stable Diffusion has been recognized as the best model, especially in the context of creating conceptual artwork, there is still a need to simplify the process of creating concept art and find the most suitable generative model. This research used three methods: Latent Diffusion Model, Dreambooth: fine-tuning Model, and Stable Diffusion. The research results show that the Dreambooth model produces a more real and realistic painting style, while Textual Inversion tends towards a fantasy and cartoonist style. Although the effectiveness of both is relatively high, with minimal differences, the Dreambooth model is proven to be more effective based on the consistency of FID, PSNR, and visual perception scores. The Dreambooth model is more efficient in training time, even though it requires more memory, while the inference time for both is relatively similar. This research makes a significant contribution to the development of artificial intelligence in the creative industries, opens up opportunities to improve the use of generative models in creating conceptual works of art, and can potentially drive positive change in the use of artificial intelligence in the creative industries more broadly.

**Keywords:** *text-to-image generation; fine-tuning model; dreambooth; textual inversion, automatic1111;*

## 1. INTRODUCTION

Image synthesis converts text, sketches, or other sources into images using a generative artificial intelligence model [1]. Generative models are the most influential class of generative models in artificial intelligence thanks to their ability to generate data. The major achievements of this model are in the fields of computer vision, speech generation, bioinformatics, and other areas of natural language processing that are currently being developed [2]. There are four main models that are widely used, namely, Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), Normalizing Flows, and Diffusion models [3].

In creating synthetic images, several generative models compete to obtain high effectiveness and efficiency, such as the Generative Adversarial Network (GAN), which consistently, from year to year, can become a top-level model in creating works of art in the form of images, music, and literature [4]. However, GAN's steps were stopped after research conducted by Dhariwal P and Nichol A [5], from OpenAI, which stated that the diffusion model beat GAN in creating synthetic images. This research explained that the Generative Adversarial Network (GAN) had shortcomings in the architecture and image results produced by the model. The GAN architecture performs computations that are much more complicated and take longer in terms of time and memory usage in running the model, making GANs inferior to diffusion models in effectiveness. Likewise with efficiency, the image quality resolution results produced by the diffusion model are much better than the Generative Adversarial Network (GAN) as evidenced by the results of the FID (Fréchet inception distance) score in this study, a score of 3.85 with a resolution of 512×512 on the diffusion model, and GAN with a score of 7.72 at a resolution of 512×512, the lowest score is the best.

Over time, various developers have modified and developed the diffusion model. According to research conducted by regarding the development of the text-to-image diffusion model, the diffusion model experienced very significant development, starting from OpenAI, which released the Dall-e and Dall-e 2 models, then Google, which released Imagen, and StabilityAi, which released Stable Diffusion [6]. These models have the same basis, namely the diffusion model. Robin Rombach conducted research comparing these models, from the models mentioned earlier [7] stated that Stable Diffusion has a higher level of effectiveness and efficiency than other models. This is proven by the FID (Fréchet inception distance) score [8] of each model, which states that stable diffusion is better than that of the other models.

Stable Diffusion also has limitations [9], one of which is that it cannot create specific images and provide personalization, the Stable Diffusion model can still produce artifacts, such as blurring, checkered patterns, or color inconsistencies, especially in somebody's anatomy including faces and scenes. Complex. These artifacts can affect the overall visual quality of the resulting image. The Stable Diffusion problem was answered by research conducted by Ruiz [10], namely, fine-tuning a model called Dreambooth. This fine-tuning model provides a solution for Stable Diffusion to create more varied and clear synthetic images. Research conducted by Gal [11], also has the same solution for Stable Diffusion, namely fine-tuning the Textual Inversion model, which provides the same solution as the Dreambooth model. The difference between these models lies in their respective architectures. Dreambooth has a more complex architecture that produces larger model data than Textual Inversion. Meanwhile, Textual Inversion will focus on text embedding, which makes the Textual Inversion model much smaller than Dreambooth. This comparison aims to compare the effectiveness and efficiency results of the fine-tuning model applied to Dreambooth and textual Inversion.

## 2. METHODS

### 2.1. Latent Diffusion Model

Diffusion models are a new generative model class that produces various high-resolution images. This model attracted a lot of attention after OpenAI, Nvidia, and Google carried out large-scale data training on this model. Other examples of architectures that use diffusion models are GLIDE, DALLE-2, and Imagen, and only Stable Diffusion is open-source (How Diffusion Models Work: The Math from Scratch | AI Summer, n.d.). Diffusion models are basically very different from previous generative models. Intuitively, the diffusion model aims to decompose the process of image creation (sampling) into even smaller "denoising" steps. The intuition behind this is that the model can improve itself through small steps and gradually produce very good samples. Basically, the model has been applied to the alpha fold model, but the iteration process makes the model slower in taking samples when compared to GAN (Ho et al., 2022)

In more detail (Rombach et al., 2022) explained in their research that they suggested adding an encoder network to input into latent space such as z_t=g 〖(x〗 _t). The intuition behind this decision aims to minimize the computational demands on the diffusion model by processing the input data in a smaller dimensional space. After that, the basic diffusion model is applied with the addition of UNet Architecture to generate new data, which is carried out by the decoder network.
The following losses are of the same type as the diffusion model (DM) formulated as follows:

$$L_{DM} = \mathbb{E}_{x,t,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(x_{t,}t)\right\|^2\right] \qquad (1)$$

The following loss Latent diffusion model (LDM) is formulated as follows:

$$L_{LDM} = \mathbb{E}_{\varepsilon\,(x),t,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(z_{t,}t)\right\|^2\right] \qquad (2)$$

Basically, the difference between the diffusion model and the latent diffusion model is only in the encoder, which is notated as ε, and changing the image variable x to z. A detailed explanation of the latent diffusion formula is as follows:
$\mathbb{E}_{\varepsilon\,(x),t,\epsilon}$ = s is the sample image notation, ε is the encoder, t is the timesteps, ϵ is the noise.
$\epsilon - \epsilon_\theta(z_{t,}t)$ = *the UNet architecture formula.*

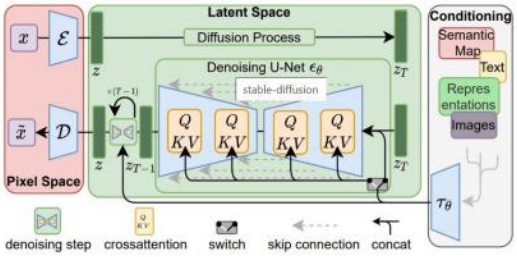

**Figure 1.** *Overall latent diffusion model architecture*

The basic method used is the Latent diffusion model (LDM). The latent diffusion model carries out the diffusion process in latent space, which means that this model consumes less training costs and has a faster inference time [7]. Observations showed that the image fragments retained perceptual detail and conceptual semantic composition after the image underwent fairly high compression. LDM provides noise in perceptual and semantic compression using a generative process that first learns the trimming (cutting) stages of pixel-level redundancy using an autoencoder and then manipulates or generates semantic concepts with a diffusion process [8].

The latent diffusion model consists of two core components: the autoencoder and the diffusion model [12]. Autoencoder is a collection of pre-trained image data with a large capacity. The autoencoder learns how to convert an image into a spatial latent code using KL-divergence loss and vector quantization regulations. Then, a decoder learns how to convert the latent map into a suitable image. The second component in LDM is a diffusion model, trained to produce code using latent space. This diffusion model can be conditioned on class labels, segmentation masks, or a text-embedding model that has been trained [13].

## 2.2. Dreambooth Model

The dreambooth model provides personalization to a text-to-image model, such as Stable Diffusion with input samples of 3 – 5 image data that suit the subject. This makes it possible to create an image subject that has different contextual features in certain scenes, poses, and viewpoints. With the help of a unique identifier, this method succeeded in embedding an example subject into the output domain created [9], [10].

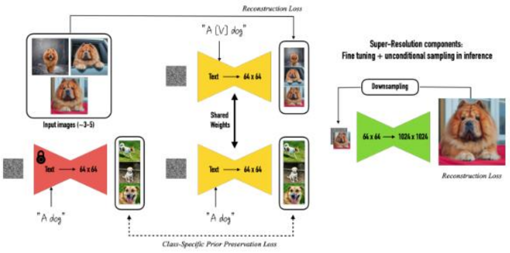

**Figure 2.** *Dreambooth model fine-tuning algorithm*

The basic architecture of a Dreambooth model method, with the following steps :

a. Input: there are two types of input data. The first is image training samples or images of specific objects with a certain number, such as 3 or more images that have the same specifications. Example of a "corgi" image with a total of 5 images by entering the same "corgi" image. On the other hand, the input entered is a sentence where each word will be converted into a number called a vector, for example, "a" has its own value, as does "photo", "of", and "sks" each word will be converted into certain numbers. The input sentence is called a unique identifier, directly related to the input image.

b. Add Noises: the noise given to the image sample is divided into two parts. One part is used to carry out the diffusion process, and the second part is used to compare the output results of the diffusion process. Please note that adding noise to an image causes the image to become undefined. The division of the two parts, for example, the first part is given noise x(n), then the second part is noise x(n-1) as a comparison for the output diffusion process [13].

c. The diffusion process is used in a fine-tuning model using stable diffusion, namely the text-to-image model, which is the basis of the fine-tuning method. Stable diffusion has the same basis as the latent diffusion model, with each diffusion model having forward and reverse steps. The forward step is the stage of adding noise to an image, and the reverse step is the stage of denoising or reducing noise in an image, which is called a Markov Chain [8], [13].

d. Gradient Update is a punishment or punishment stage. This term is given to the results or output images produced from the diffusion process, which, after being compared with sample data, still have a high loss (bad prediction). If the resulting loss is high, the output will repeat the diffusion process so that the image can produce an image that matches the existing sample data with the iteration input limits entered by the user.

e. Output will produce an image that resembles the sample image combined with a unique identifier to produce an image with certain personalization such as the scene, pose and viewpoint of an image.

### 2.3. Textual Inversion

Textual Inversion is a method that provides freedom to be creative through a text-to-image model by providing 3-5 images that represent the user's desires, such as an object or the painting style of an image. This method studies data through "words" in the embedding space of a text-to-image model. These "words" can be natural language sentences or sentences commonly used in everyday life by combining specific personalization of objects intuitively [11].

The architecture and working process of Textual Inversion are not too different. Only one process is different at the Gradient Update stage. Dreambooth provides punishment for objects that have high losses to be returned to the diffusion process, but Textual Inversion changes the vector of unique identifier words so that the system learns to get the right words to produce output that matches the sample data. This process produces a text embedding that can be used as a trigger prompt or word to produce the desired image.

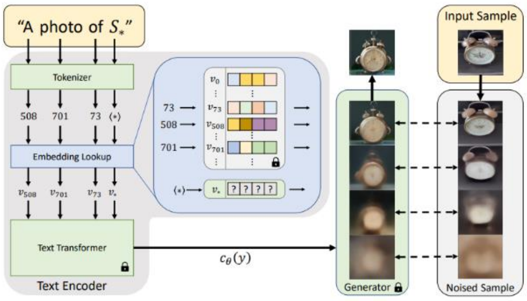

**Figure 3.** *Textual inversion algorithm*

In the text encoding stage of most text-to-image models, the first stage involves converting the input text into a numerical representation. This is usually done by converting words into tokens, where each token is equivalent to an entry in the model dictionary. Then, those entries are converted into "embeddings" – continuous vector representations for specific tokens. Typically, these embeddings are learned as part of the training process. In our research, we discovered new embeddings that represent specific visual concepts provided by users. This embedding is then associated with a new pseudo-word, which can be included in a new sentence like any other word.

## 3. RESULTS AND DISCUSSION

The experiment went through several stages and limitations, including the hardware used. The Dreambooth model and Textual Inversion have a system requirement recommendation of at least 16 GB VRAM on the GPU, while we only use 12GB VRAM on the GPU. In deep learning, the GPU or graphics card plays an important role in running the system. Stability AI, one of the companies that makes the Stable Diffusion model, uses around 256 A 100 GPUs, with an estimated price of one GPU at $200,000 US dollars [14].

Data training was carried out using webUi Automatic1111 to make it easier to produce an image. In the training stage, we combine a private dataset with one of the Stable Diffusion models. The experiment used the default settings on automati1111 with changes to the steps, namely 10, 20, 30, 40, and 50 steps [15]. The Dreambooth model can train 140 epochs with 13 steps for 1820 steps, and Textual Inversion can train 200 epochs with 20 steps for a total of 4000 steps.

Each method carries out experiments at different steps, and each step produces 50 images according to the dataset created so that the measurements are not biased or balanced. The total number of experimental images produced was 500, divided into 250 Dreambooth models and 250 Textual Inversion images with each step of 10, 20, 30, 40, and 50, totaling 50 images.

### 3.1. Dataset

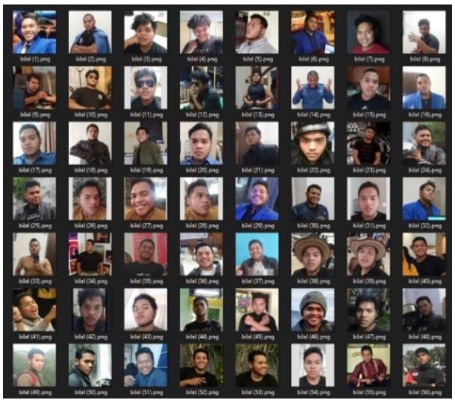

***Figure 4**. Dataset*

The process of creating a dataset aims to provide personalization to images. The image used is the face of the first author (Bilal Abdul Qowy) in this study. A total of 50 images with a resolution of 512 x 512 pixels, each with different expressions, backgrounds, clothes, and hairstyles. Added descriptive sentences that describe one image in the dataset created. These descriptive sentences will later be used in training data for Textual Inversion to recognize images that will be entered into Textual Inversion to produce a new text embedding. This research uses images as the main data. Image data can be used to extract the qualitative information contained therein. In the context of image analysis, qualitative features such as color, shape, texture, or object can be extracted to provide an understanding of the visual characteristics in the image.

### 3.2. Effectiveness

Effectiveness is measured using 3 indicators: Fidelity, Diversity, and Image quality.

### 3.2.1. Fidelity

Image fidelity refers to how accurate the image produced by a model is without excessive distortion or loss [8], [16]. The Fidelity value of an image can be measured using Fréchet inception distance (FID) with the help of human visual perception, namely direct assessment by human perception. The measurement uses the FID score [8], [17] by comparing the two datasets, namely the original facial image dataset (real image) and the dataset created by a generative model (generated image). The FID score measures the distance between the similarity or suitability of the original data to the generative data. Therefore the smaller the score produced, the better the model is at creating generative images.

**Table 1.** *Table of dreambooth results*

| Steps N | FIDs Dreambooth | FIDs Textual Inversion |
|---|---|---|
| 10 | 7.2 | 7.3 |
| 20 | 2.8 | 4.8 |
| 30 | 2.2 | 3.9 |
| 40 | 3.5 | 3.5 |
| 50 | 2.4 | 3.0 |

The results from Dreambooth are not very consistent because there are increases and decreases in scores in certain steps. However,

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i1.37942

the score produced by Dreambooth is far superior to that of Textual Inversion. Dreambooth can reach 2.2. The FID score shows that the resulting image is better than Textual Inversion. The score from Textual Inversion shows good consistency from the initial steps to the limit of the steps being tested. Even though it is consistent, the resulting score is only up to 3.0, indicating that the results from Textual Inversion are not better than the Dreambooth score.

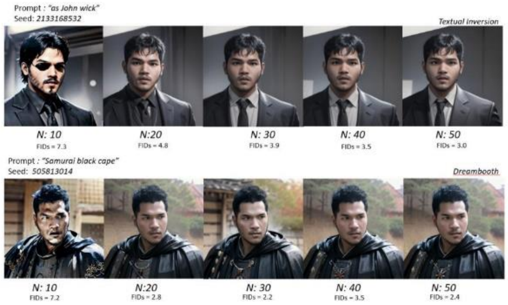

**Figure 5.** *Comparison of the fidelity of the two models*

Determining the effectiveness of an image requires more than just an FID score. Human visual perception plays an important role in determining the effectiveness of an image. In Figure 4.3, the first 10 steps of the Textual Inversion model show a score of 7.3, and the results are very ineffective because there are still distortions on the face and the texture is too rough. Moving on to the next steps, 20 shows a clear increase in the image on the face but with relatively the same background and not too detailed. Facial detail increases in the next step, 30 steps, followed by a change in the color of the object's clothing. Facial detail continues to increase in the next steps, 40 steps add detail to the face with better color tones. In the last experiment, 50 steps, details become clearer with increasing lighting and composition according to the image object.

The Dreambooth model creates images with a more real and realistic style. Therefore, Dreambooth's FID score is lower than that of Textual Inversion. The FID score measures the distance of similarity between the two datasets being compared, therefore, Dreambooth has a lower score than Textual Inversion. However, that doesn't mean that Dreambooth is much more effective than Textual Inversion. On the first try of 10 steps, the resulting image still looks very rough, the texture is messy, and the color contrast is very high. In the next 20 steps, the image has improved quite significantly. The face looks clear, and the facial texture is more detailed. In the 30-step experiment, facial details looked better, and the background of the object changed. At 40 steps, the face again looks a little blurry but has a face shape that more closely resembles the training data. In the last trial of 50 steps, the face changed slightly in lighting and coloring.

The only striking difference is in step 10. The next steps show differences in small details in the image.

3.2.2. Diversity

Diversity image refers to the diversity results produced by a particular model. Measuring diversity does not have an exact measurement. Diversity image assessment uses human visual perception [18] but can also be measured through the FID score. The FID score results are already known in the Fidelity experiment, so this stage only focuses on human visual perception. The following are the results of each model's fine-tuning method.

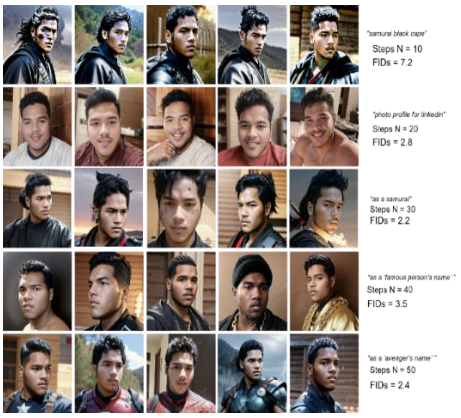

**Figure 6**. *Dreambooth diversity model*

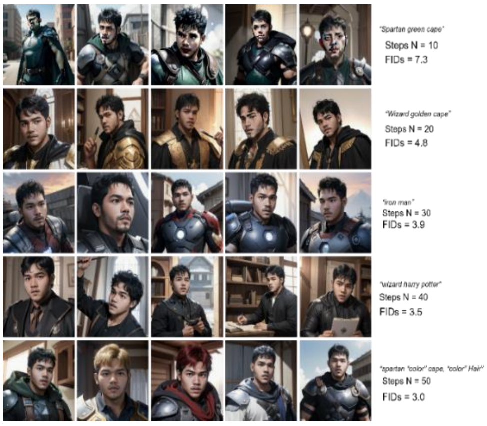

**Figure 7**. *Textual inversion diversity*

The results of Textual Inversion produce more cartoonish images. Having a higher FID score than Dreambooth does not mean that the image produced by Textual Inversion is worse than the Dreambooth model. This model is very relevant in creating fantasy-themed images.

### 3.2.3. Image Quality

The image quality of an image is measured using peak signal-to-noise ratio (PSNR) [19], [20]. The PSNR method measures two distances between the original and compressed images. Images that have a high value mean they have better quality.

**Table 2** *Comparison of PSNR scores of the two models*

| N | PSNRs Dreambooth | PSNRs Textual Inversion |
|---|---|---|
| 10 | 44.72 | 41.52 |
| 20 | 45.58 | 42.44 |
| 30 | 45.31 | 43.81 |
| 40 | 45.77 | 44.60 |
| 50 | 45.59 | 44.21 |

The results of each model show that both have good image quality, but if you compare the PSNR scores on the two models, it can be concluded that the Dreambooth model has a higher score than Textual Inversion. Therefore, the image quality that has a higher level of effectiveness is the Dreambooth. Model. However, we can underline that the Dreambooth model and Textual Inversion results have different image painting styles.

### 3.2.4. Efficiency

Efficiency is measured using 3 indicators: training time, inference time, and memory usage.

### 3.2.5. Training Time

**Table 3**. *Training time comparison table*

| | Dreambooth Model | | Textual Inversion | |
|---|---|---|---|---|
| Settings | *Epoch* | *Steps* | *Epoch* | *Steps* |
| | 140 | 13 | 200 | 20 |
| Training time | 00:29:28 | | 02:21:00 | |

Both models have different default settings on the system, the time taken is quite significant. Dreambooth is 3 times faster than Textual Inversion; the biggest possibility of a fairly high difference in training time is in the default settings for training the two models. Issues arise when the Dreambooth model is forced to train more than the specified epoch, such as 'ran out of GPU memory' where the GPU can no longer accommodate the memory used.

### 3.2.6. Inference time

Inference time is the time a model takes to create or create a synthetic image. In this experiment, the author took samples of 50 images from each step. 10 steps to 50 steps. The average time taken in the first 10 steps is 3 seconds or seconds. Increased to 5 seconds, the next step also increased to 6 seconds, and so on, up to 9 – 10. The difference between the two models, the Dreambooth model, and the Textual Inversion model, is not very significant because the difference between the two models is under 1 second. The Dreambooth model has a faster travel time of around 0.40 seconds.

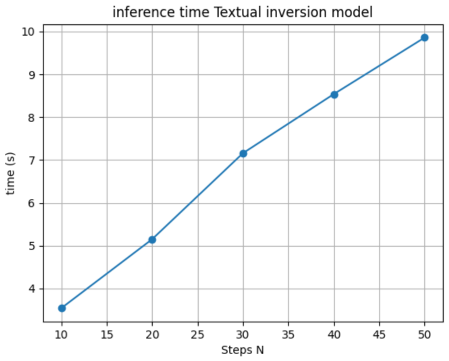

**Figure 7.** *Inference time Dreambooth model graph*

In the first 10 steps, the average time taken is 3 seconds. Increased to 5 seconds, the next step also increased to 6 seconds, and so on up to 9 – 10 seconds. The difference between the two models, the Dreambooth model, and the Textual Inversion model, is not very significant because the difference between the two models is under 1 second. The Dreambooth model has a faster travel time of around 0.40 seconds.

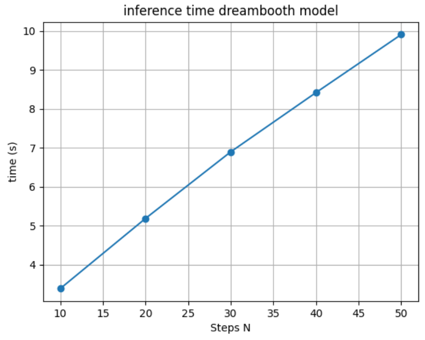

**Figure 8.** *Inference time Dreambooth model graph.*

**Table 4**. *Table of Dreambooth Results*

| Step N | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| 1 | 4.07s | 5.29s | 6.80s | 8.52s | 10.36s |
| 2 | 3.42s | 5.19s | 6.82s | 8.58s | 9.85s |
| 3 | 3.21s | 5.15s | 7.06s | 8.49s | 9.95s |
| 4 | 3.21s | 5.27s | 6.89s | 8.45s | 9.94s |
| 5 | 3.30s | 5.19s | 6.94s | 8.42s | 9.95s |
| 6 | 3.04s | 5.21s | 7.58s | 8.34s | 9.82s |
| 7 | 3.33s | 5.20s | 6.92s | 8.15s | 10.01s |
| 8 | 3.14s | 5.13s | 6.83s | 8.56s | 9.79s |
| 9 | 3.39s | 5.13s | 6.64s | 8.33s | 9.69s |
| 10 | 3.28s | 5.23s | 6.54s | 8.41s | 9.79s |
| mean | 3.39s | 5.19s | 6.90s | 8.42s | 9.91s |

**Table 5**. *Table of Textual Inversion Results*

| Step N | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| 1 | 3.23s | 5.16s | 6.88s | 8.51s | 9.69s |
| 2 | 3.33s | 5.16s | 6.97s | 8.44s | 9.64s |
| 3 | 3.32s | 5.14s | 6.85s | 8.50s | 9.87s |
| 4 | 3.22s | 5.12s | 6.69s | 8.69s | 10.03s |
| 5 | 3.32s | 5.01s | 6.91s | 8.38s | 9.97s |
| 6 | 3.58s | 5.39s | 6.59s | 8.49s | 9.57s |
| 7 | 3.85s | 5.16s | 6.86s | 8.42s | 9.72s |
| 8 | 3.81s | 5.31s | 6.86s | 8.64s | 10.29s |
| 9 | 3.95s | 5.10s | 6.84s | 8.52s | 9.97s |
| 10 | 3.84s | 4.95s | 7.11s | 8.52s | 9.85s |
| mean | 3.54s | 5.15s | 7.16s | 8.54s | 9.86s |

### 3.2.7. Memory usage

In the average inference time experiment to create an image, 7GB VRAM is needed on the Dreambooth model, or around 57.64% of the total memory on the hardware used. In MSI afterburner memory usage during training, the two 67 Inversion models use less VRAM, with an average total memory of 5GB VRAM, or 43.19% of the total memory on the hardware. This comparison makes the Textual Inversion model a more efficient model with a difference of 14.45%.

In training data, the Dreambooth model requires 6 - 12GB VRAM on the GPU system, which means it uses almost 100% of the GPU memory capacity, while Textual Inversion requires 6 - 8GB GPU memory, which means it only uses around 75% of the total GPU memory.

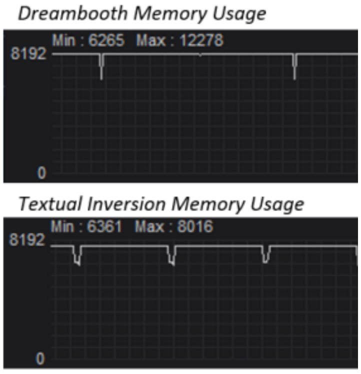

**Figure 6.** *MSI afterburner memory usage when training both models*

The data results produced in the two models are quite different. The Dreambooth model produces a data file of 5.21 GB with 140 files. Textual Inversion produces only 68.8MB with 242 files. This difference is because the Dreambooth model produces a new model file in the "ckpt" format, which is a format similar to creating a new model. Textual Inversion only produces text embedding files with contents in the form of a collection of texts that have been trained.

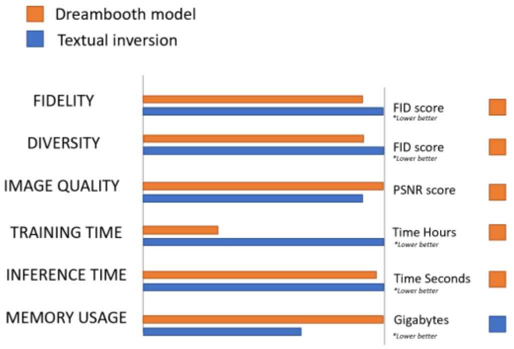

**Figure 7**. *Dataset*

From the results of 6 comparison indicators, Dreambooth beats textual Inversion in 5 indicators, and textual Inversion can beat Dreambooth only in 1 indicator, namely memory usage. It can be concluded that the Dreambooth model is more effective and efficient in creating synthetic images.

## CONCLUSION

It can be concluded that the Dreambooth model dominates every measurement indicator except memory usage because the memory requirements for training a Dreambooth model require quite a large amount of memory. This doesn't mean that Textual Inversion is a bad method; the difference between the values of Dreambooth and Textual Inversion is not that different, but just the implementation and way of using it are different. The results of the two also have different painting styles. The benefit obtained from this experiment is that the results of the tests carried out can be used as a reference for further research in the field of machine learning or deep learning.

## REFERENCES

[1] S. S. Baraheem, T. N. Le, and T. V. Nguyen, "Image synthesis: a review of methods, datasets, evaluation metrics, and future outlook," *Artif Intell Rev*, vol. 56, no. 10, pp. 10813–10865, Oct. 2023, doi: 10.1007/s10462-023-10434-2.

[2] H. Cao *et al.*, "A Survey on Generative Diffusion Model," Sep. 2022, [Online]. Available: http://arxiv.org/abs/2209.02646

[3] C. Liu *et al.*, "Generative Diffusion Models on Graphs: Methods and Applications," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.02591

[4] S. Shahriar, "GAN Computers Generate Arts? A Survey on Visual Arts, Music, and Literary Text Generation using Generative Adversarial Network."

[5] P. Dhariwal, ⇤ Openai, and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis."

[6] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, "Text-to-image Diffusion Models in Generative AI: A Survey," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.07909

[7] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models." [Online]. Available: https://github.com/CompVis/latent-diffusion

[8] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded Diffusion Models for High Fidelity Image Generation Figure 1: A cascaded diffusion model comprising a base model and two super-resolution models. *. Equal contribution," 2022.

[9] "How to Fine-tune Stable Diffusion using Dreambooth," https://towardsdatascience.com/how-to-fine-tune-stable-diffusion-using-dreambooth-dfa6694524ae. Accessed: May 10, 2023. [Online]. Available: https://towardsdatascience.com/how-to-fine-tune-stable-diffusion-using-dreambooth-dfa6694524ae

[10] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation." [Online]. Available: https://dreambooth.github.io/

[11] R. Gal *et al.*, "An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion," Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.01618

[12] Lilian Weng, "What are Diffusion Models?" Accessed: May 10, 2023. [Online]. Available: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models." [Online]. Available: https://github.com/hojonathanho/diffusion.

[14] "Stability AI Stable Diffusion Public Release." Accessed: May 10, 2023. [Online]. Available: https://stability.ai/news/stable-diffusion-public-release

[15] "Stable Diffusion WebUI AUTOMATIC1111: A Beginner's Guide." Accessed: May 10, 2024. [Online]. Available: https://stable-diffusion-art.com/automatic1111/

[16] E. Hoogeboom, E. Agustsson, F. Mentzer, L. Versari, G. Toderici, and L. Theis, "High-Fidelity Image Compression with Score-based Generative Models," May 2023, [Online]. Available: http://arxiv.org/abs/2305.18231

[17] "How to Implement the Frechet Inception Distance (FID) for Evaluating GANs," Oct. 2019, Accessed: May 10, 2023. [Online]. Available: https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/

[18] A. Mcnamara, "Visual Perception in Realistic Image Synthesis," 2001.

[19] "Python | Peak Signal-to-Noise Ratio (PSNR)." Accessed: May 10, 2023. [Online]. Available: https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/

[20] F. A. Fardo, V. H. Conforto, F. C. De Oliveira, and P. S. Rodrigues, "A Formal Evaluation of PSNR as Quality Measurement Parameter for Image Segmentation Algorithms."