Protect Art and Creativity: A Prevention Framework for Unauthorized Learning of Text to Image AIs.

Jinho Kim[1, 2, †, §], Jooney Han[1, †]
jhk@zippercorp.com, jhan756k@gmail.com
[1]Korean Minjok Leadership Academy, Hoengseong-gun, Gangwon-do, Korea
[2]Zipper AI Research Lab
†: Equal Contribution §: Corresponding Author

Abstract In this work, we aim to solve the problem of unauthorized learning of works arising from the process of collecting large amounts of data from Text to Image (TTI) AI models represented by Stable Diffusion. The TTI model performs indiscriminate web data crawling to collect a substantial number of images, and these images are used for model learning without the consent of the original author. The TTI model is capable of learning the drawing style of an image, which undermines the value of the original work. Therefore, we suggest a method of transforming images to deteriorate the learning accuracy of TTI models. Then, we compare the quality of original images to images processed by the modification method presented in this study, using both quantitative measurement and qualitative measurement. Thus, we confirm that the image modification method we propose prevents AI models from learning literary works without permission.

1. Introduction

1.1 Background of Study

After the publication of *Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models."*, in 2022, various Text to Image AIs have been distributed.[1] For example, instead of using the existing GAN (Generative Adversarial Network) model, Stable AI developed a new Diffusion model called Stable Diffusion AI, which was trained for approximately 150,000 GPU hours. The entire model and source code are set public for free, and it outperforms existing paid image generation models such as DALL-E. [2] Stable Diffusion AI also provides a method to create a user-defined model, called Fine-Tuning, under a certain license. Several AI models adopting the model have been introduced. However, in the learning process of these models, copyright issues have been raised as the models are trained with images without the creator's consent. For instance, Novel AI image generator, a paid image generation model that is optimized for generating anime-style images, trained its model using images from an illustration archive database Danbooru without the original author's consent. Thus, considering the need for image copyright protection against AI, we aimed to protect the rights of creators who do not want their creations to be trained by AI image generation models by developing a simple yet effective method.

1.2 Prior Work

Research conducted on Text to Image AI has been rapidly increasing since 2022, and the GAN method of image generation since 2014. In *Goodfellow, Ian J., et al. "Generative adversarial networks." arXiv preprint arXiv: 1406.2661 (2014).*, the GAN model is mathematically designed and implemented, and the experimental generation model is successfully built using MNIST, TFD, and CIFAR-10 datasets.[3] Later, improved versions of the GAN model, in the order of DCGAN, PG-GAN, BigGAN, and StyleGAN, have been developed, improving the model's overall performance greatly.[4-7] However, problems such as unstable training, easy overfitting, and lacking diversity of data were raised, later leading to the development of the Latent Diffusion model.[7-8]

In the process of training these models, the method for pruning images that may be inappropriate, illegal, or cause overfitting has been developed. However, the method to protect literary works from being used without permission has not been researched previously. We think this is mainly because the Text to Image AI model has only started to be used recently along with the publication of the Stable Diffusion model, while the ongoing research mainly focuses on the technical sides of the model, not on the side effects of it.

2. Theoretical Background

2.1 Analysis of Text to Image AI's Image Generation Mechanism

2.1.1 Text to Image AI Based on GAN Model

The GAN (Generative Adversarial Network) model is an AI model that is used extensively in the field of Text to Image AI. The GAN model can be divided into two parts: the Generator(G) and the Discriminator(D). During the training process, the Generator and the Discriminator acts as adversaries to each other, as D distinguishes between the fake sample G generates and the real sample from the training dataset and outputs the percentage of the similarity of the output sample and a real sample as a number between 0 and 1. G receives a vector composed of random numbers as an input and outputs a generated sample that is created using the vector values. D outputs the score of the sample, which is then used by G through backpropagation. As this is repeated, the quality of the sample G generates increases, and G possesses the ability to generate a sample that is indistinguishable from a real sample.[4]

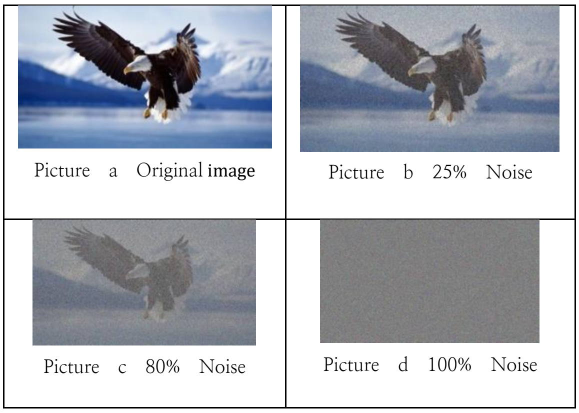
In this process, since D evaluates only true or false, a binary classification function for a loss function is used. In this research, we used the Binary Cross Entropy (BCE), which is shown mathematically below.

$$BCE(x) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\big(h(x_i;\theta)\big) + (1-y_i)\log(1-h(x_i;\theta)) * -$$

The GAN model has been improving ever since the initial model was developed. DCGAN, which uses a deep convolutional neural network, BigGAN, which utilizes large-scale training, and StyleGAN, which is a style-based generator architecture that is specialized in changing style components in an image. However, image generation based on the GAN model encountered several problems, such as high overfitting probability during the training process, restriction when creating a completely new image, and high hardware resource cost. [4-7]

2.1.2 Text to Image AI Based on Latent Diffusion Model

The Latent Diffusion model was proposed after the GAN model, fixing the GAN model's downsides with improved generation quality. It is an image generation model developed by the collaboration of the LMU University of Munich and Heidelberg Scientific Computing Center in 2022. Latent Diffusion and the GAN model have very different training processes. The first step of the Latent Diffusion model's training process is to add noises to the image dataset. The images below show how the noises are added during the process.


Picture a Original image

Picture b 25% Noise

Picture c 80% Noise

Picture d 100% Noise

Figure 1 The process of adding noise to an image

Like [Figure 1], the latent diffusion model adds noise to an image linearly and uses this data for the training process. Then, the model is given information about the original image, and is asked to reconstruct the image in reverse order. Starting from images with relatively less noise, the model is gradually fed with more and more noise (using Gaussian Noise, described later [10]) until image that is 100% noise is obtained. Through this procedure, the model acquires an ability to generate a new image from complete noise without any visual information.

Also, apart from the existing Diffusion model, Latent Diffusion model involves the Latent Space Encoding process. During the Latent Space Encoding process, parts of an image with different characteristics are divided into segments and are expressed as matrices. The image below shows the image of a sunset that was processed with Latent Space Encoding.

Figure 2 a. original image of a sunset

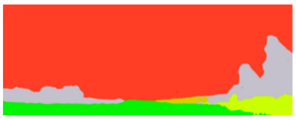


Figure 2 b. sunset image after the encoding process

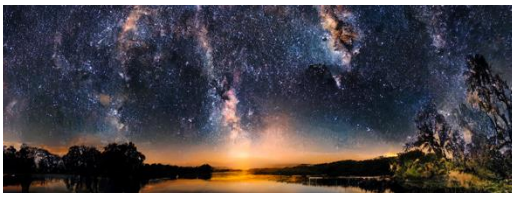


Figure 2 c. Image generated based on the encoded image

LAION-5B Dataset was used for the training process of the Stable Diffusion model. It is composed of over 5.8 billion CLIP data, which is a set of image and text. [11] Before the actual usage of the model, we inspected the dataset and gained insights on the types of data in the dataset, using LAION's backend URL.

As described above, we confirmed that the LAION-5B dataset uses various data from the web by crawling them in an indiscreet way. Besides individual artist's cat illustrations, informal artworks such as memes and completely irrelevant Text CLIPs, such as website links could also be found. These examples prove the point that LAION-5B saves all images uploaded to the web during the data collection process.

Also, Fine-Tuning of the Latent Diffusion model allows users to additionally train another object, drawing style, etc. for their own needs. However, as this is third-party, platforms may create image generation services by Fine-Tuning cartoons or online illustrations, all the while ignoring the consent of illustrators.

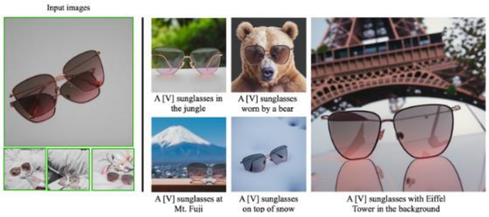

Input images
A [V] sunglasses in the jungle
A [V] sunglasses worn by a bear
A [V] sunglasses at Mt. Fuji
A [V] sunglasses on top of snow
A [V] sunglasses with Eiffel Tower in the background

Figure 5 Fine-Tuning an image of sunglasses

Figure 3 Data obtained with keyword "cat memes"

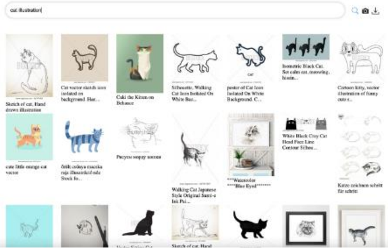


Figure 4 Data obtained with keyword "Cat illustrations"

The implementation of features and methods of Latent Diffusion mentioned above is distributed online by the name of Stable Diffusion. Supported and maintained by StableAI and LAION, it is currently the most used implementation of the Latent Diffusion model.

For this research, we used the Stable Diffusion model to experiment with our method of image modification. This is because it is proven very powerful performance-wise and is open-source. We concluded that using the GAN model is inappropriate for our research since it has too many derived methods and doesn't have a fixed dataset for training. The high requirement of hardware resources for Fine-Tuning was also a problem.

3. Experimentation methods

3.1 Designing the method to prevent Stable Diffusion model from training unauthorized data

3.1.1 Usage of Stable Diffusion and DreamBooth

The Stable Diffusion method is completely open source. In this research, for the sake of efficiency during the experiment process and flexibility of manipulating parameters during the Fine-Tuning process, we used the Stable Diffusion WebUI.

Also, we made use of the DreamBooth technique for Fine-Tuning. Presented in *Ruiz, Nataniel, et al. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation." arXiv preprint arXiv:2208. 12242 (2022).* DreamBooth enables us to fine-tune a Stable Diffusion model with relatively small set of additional data. [12]

In [Figure 5], 4 different images of the same sunglasses are given a class name [*V*]. It can be used in a prompt to generate an image like the following way: 'A [*V*] sunglasses in the jungle'. Thus, using DreamBooth for Fine-Tuning, in this research, we modify additional data and confirm whether the model can learn the artistic style, drawing style, or elements within illustrations and generate meaningful images.

3.1.2 Decrease in learning rate due to image modification

As mentioned above, this study aims to prevent unauthorized learning of images using two main methods. Firstly, we propose a method to prevent learning by utilizing the characteristics of Stable Diffusion. In this process, while authorized users can perceive the images without difficulty, the AI model undergoes difficulties due to the image modification during the learning process. To achieve this, we have developed various hypotheses and corresponding image modification programs.

First, we conducted research on the process by which humans perceive images. According to previous studies, when disappearance or generation effects are applied to images, objects are perceived through atypical gaze. Additionally, when the contrast of colors is increased, illusions are induced, or when encountering unfamiliar objects or artworks in an art gallery, people consciously examine the images in detail by scanning the entire image. On the other hand, when recognizing a human face without significant awkwardness, the brain unconsciously abandons detailed recognition processes and focuses the gaze on the forehead to recognize the person. [13-17] In the case of digital illustrations addressed in this study, according to previous research, unlike artworks, humans do not consciously examine them in detail but rather perceive objects roughly by gazing at the forehead, as mentioned before. Taking this into consideration, the researchers aimed to minimize the difficulty for humans to recognize characters and objects in illustrations while modifying the images to protect them from being properly learned by the Stable Diffusion model.

Therefore, we devised a method to modify the images and evaluated how much the modified images reduced the learning efficiency of the Stable Diffusion model. The following methods were used to modify the images.

To disable smooth Latent Space Encoding, the edges of objects in the images are modified. By recognizing the edges of objects and modifying them, such as expanding or thickening specific parts of the edges, the encoding process becomes more challenging.

To recognize the edges of the images, the Canny Edge Detection algorithm was employed. Among the edge detection algorithms that distinguish corners, the Canny Edge Detection algorithm uses a Gaussian filter to eliminate any noise that interferes with corner detection. Then, the overall intensity of the image is calculated, and points where intensity is changed abruptly are identified as edges based on minimum and maximum thresholds. [18-19]

After confirming the element-specific edges of the images through Canny Edge Detection, the thickness of the edges was randomly modified or deleted through a series of processes. The obtained images were then fine-tuned and used to train the Stable Diffusion model to check for any awkwardness in the image's reconstruction.

To implement a program that applies a blur effect, we conducted research on methods for applying a blur effect. Generally, a blur effect calculates the average value of the surrounding N×N pixels when a specific parameter N is given for a pixel. In this program. To recognize the edges of objects and apply a blur effect around those edges, two variables were introduced and managed in this process. First, the "kernelSize" variable was introduced. The kernelSize variable determines how many pixels' values are included in the average calculation process for the surrounding N×N pixels of a reference pixel. As mentioned, by calculating the average value of the N×N pixels around the reference pixel and determining the value of the reference pixel, the kernelSize variable determines the range over which pixel values are calculated as the average. Increasing the kernelSize parameter value results in a smoother transformation of the image, thereby increasing the PSNR and SSIM values. Second, the "nearbyBlurSize" variable was introduced. The nearbyBlurSize variable determines how many pixels around the edges are applied with the blur effect. Therefore, increasing the nearbyBlurSize value decreases the PSNR and SSIM values. The following is part of the Python code that modifies an image by applying a blur effect to the edges of an image.

```
(…)
kernelSize = 5
nearbyBlurSize = 5

term = (kernelSize//2) + (nearbyBlurSize//2)

for x in range(term, cannyimg.shape[0]-term):
  for y in range(term, cannyimg.shape[1]-term):
    if cannyimg[x][y] == 255:
      for t in range(x-(nearbyBlurSize//2),
x+(nearbyBlurSize//2)):
       for c in range(y-(nearbyBlurSize//2),
y+(nearbyBlurSize//2)):
          for p in range(3):
            sum = 0
            for i in range(kernelSize):
              for j in range(kernelSize):
                sum += orig_img[t+i-term][c+j-term][p]
             orig_img[t][c][p] = sum //
(kernelSize*kernelSize)
(…)
```

Code 1 Code for applying blur effects to edges

Code 1 is a python code for applying blur effect to the edges of an image using the Canny Edge Detection algorithm. By adjusting the kernelSize variable, you can control the number of pixels around the boundaries that are affected by the blur effect. Increasing the kernelSize value will increase the range over which pixel values are averaged, resulting in a smoother blur effect. The following image is made with the same photo used in Figure 1 with edges recognized using Canny Edge Detection and blur effect applied.

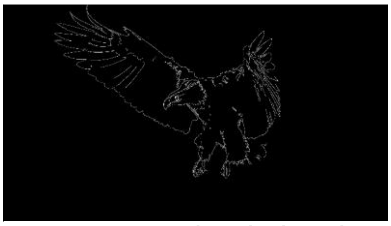


Figure 6 Image with its edge detected

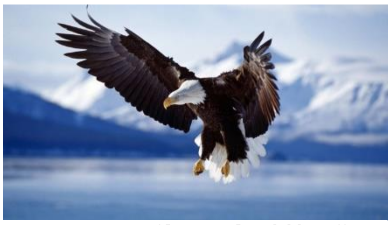
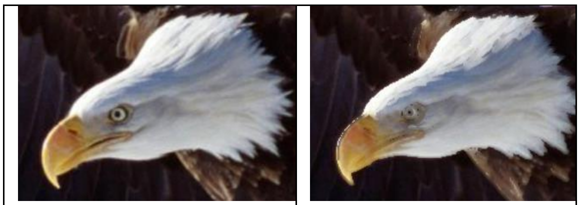


Figure 7 image of [Figure 6] with blur effect around the edges

To compare the original image with the transformed image, the researchers used qualitative analysis by visually inspecting the images. They found that the differences between the original and transformed images are not easily noticeable to the naked eye. However, when the images are zoomed in, the differences become more apparent, allowing for effective prevention of unauthorized learning by artificial intelligence models.

As you can confirm in [Figure 7], it is not easy to visually perceive the differences when the edges are modified using certain parameters. However, when the original image and the modified image are enlarged, the differences become clearly noticeable. It is through these differences that we intended to prevent unauthorized learning by artificial intelligence. Below, the original photo and the modified photo are magnified for comparison.



Picture a. magnification of the original image

Picture b. magnification of the modified image

Figure 8 Comparison of the original image and the modified image

[Figure 8], specifically [Picture b], represents the image transformed by setting kernelSize to 5 and nearbyBlurSize to 3 in Code 1. As can be observed in Figure 2 and Figure 8, we were able to qualitatively analyze that when the image is modified, it is generally difficult for users to determine whether the image has been modified in its original state, and it is challenging to perceive any awkwardness.

To quantitatively analyze the quality of the original and modified images, we introduced the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) metrics. PSNR is used to assess the level of image quality loss. It is calculated based on the Mean Squared Error (MSE) value, which measures the squared difference between the predicted value $\hat{\theta}$ and the actual value (parameter θ). In the case of images, these parameters are image pixel values. The formulas for calculating MSE and PSNR are as follows. [20-22]

$$MSE(\hat{\theta}) = E_{\theta}\left\{\left(\hat{\theta}-\theta\right)^{2}\right\} = Var_{\theta}(\hat{\theta}) + Bias_{\theta}(\hat{\theta},\theta)^{2}$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^{2}{}_{I}}{MSE}\right)$$

As can be seen from the equation, the PSNR value indicates that the image is more similar to the original photo as the MSE decreases and the PSNR value increases. Two identical images would have an infinite PSNR value.

SSIM, on the other hand, is a method used to assess visual quality differences and similarities. It utilizes the values of luminance, contrast, and structural differences. While PSNR is calculated more numerically and is data-based, SSIM considers visual elements such as luminance and contrast. [21-23]

Using the previously mentioned PSNR and SSIM values, we conducted a quantitative evaluation of the quality of the modified image. The comparison values are as follows, with three significant figures after the decimal point. The experiment was performed on a [Figure 1] image with dimensions of 1980×1080 pixels. In summarizing the results of this experiment, the degree of modification is represented as a tuple of $\mathbb{R}^{2}$, consisting of the kernelSize and nearbyBlurSize values from Code 1.

Degree of modification	(3, 3)	(5, 3)	(3, 5)	(5, 5)
PSNR (dB)	33.371	36.499	25.594	26.311
SSIM	0.988	0.993	0.957	0.960

Table 1 Image quality measurement based on the modification parameters

As can be seen from the figures and the PSNR values, it can be observed that the quality of the image varies as the degree of transformation changes, following the expected trend of increase or decrease. However, generally, it is difficult to distinguish differences in images with an SSIM value between 0.99 and 0.97, and images with an SSIM value of 0.95 or higher are considered to have acceptable quality, indicating that the level of damage is not significant, and users can perceive the image without any awkwardness. [20-23] Afterwards, to confirm the effectiveness of this modification method, we examined whether an image generation model trained on such modified images successfully generates similar-looking images.

3.1.3 Image Quality Evaluation Using G-FID Score

In this study, the goal was to prevent AI models from learning images while ensuring that ordinary users do not experience significant inconvenience in perceiving them by modifying images. To achieve this, we examined the process that humans go through when perceiving and evaluating images, as mentioned before. Subsequently, we qualitatively confirmed that the modified images do not hinder the perception process. Furthermore, for quantitative evaluation, we introduced evaluation metrics. Instead of the previously mentioned PSNR and SSIM scores, we introduced the G-FID (Generation Fréchet Inception Distance) score, which is more suitable for assessing the performance of generated images. It calculates the distance between sets of images, specifically the distance between the generated image set and the target distribution of the data to be generated. The formula for calculating the FID score is as follows, where x represents real image data, g represents generated images, μ represents the mean, and ∑ represents the covariance. [24-25]

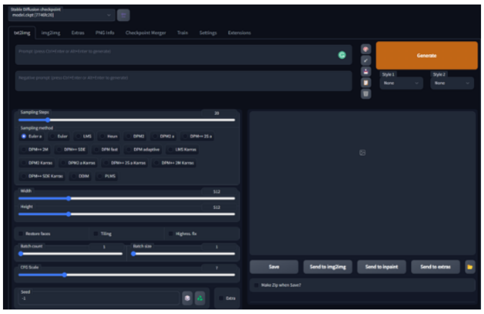
$$FID(x,g) = \left|\left|\mu_{x}-\mu_{g}\right|\right| + Tr(\textstyle\sum_{x} + \sum_{g} - 2(\sum_{x}\sum_{g})^{\frac{1}{2}})$$

In addition, we also aimed to evaluate the quality of the generated images more objectively by using the Inception Score, which is calculated using the InceptionV3 model trained on the generated image data.

4. Experimental Setup

4.1 Experiment on Preventing Image Training of AI Models via Image Modification

4.1.1 Setting up the Runtime Environment for the Stable Diffusion Model



Stable Diffusion checkpoint
model.ckpt [7460c20]

Style 1: None

Style 2: None

Sampling Steps: 20

Sampling method: ● Euler a

Width: 512

Height: 512

Batch count: 1

Batch size: 1

CFG Scale: 7

Seed: -1

Figure 9 Stable Diffusion WebUI

First, we aimed to set up an environment that enables easy usage of the Stable Diffusion technique for generating images. For this purpose, we utilized the Stable Diffusion WebUI, an open-source tool with high reliability. Additionally, to utilize Stable Diffusion in a more efficient hardware environment, we employed Google's Colaboratory Pro. The GPU used in this process was the Nvidia A100 model. The image below shows the initial screen of the Stable Diffusion WebUI executed using the Google Colaboratory platform.

To generate images using Stable Diffusion, various parameters need to be configured. In this research, we considered the time required for image generation, hardware resources, and efficiency by conducting experiments with the following settings: batch size of 1, batch count of 1, and 20 sampling steps.

4.1.2 Setting up the Fine-Tuning Runtime Environment

As mentioned earlier, the Stable Diffusion model can be fine-tuned with DreamBooth using photographs or illustrations, as mentioned in the previous paper. In order to minimize experimental errors and ensure efficient learning, we set several control variables. Firstly, for the Fine-Tuning process using DreamBooth, we used version 1.5 of the stable-diffusion model that was uploaded to HuggingFace. Additionally, instead of creating one model for the entire dataset, we generated one model for each dataset to avoid errors caused by weight variations. Furthermore, we standardized the image size. Based on previous research findings, we determined that 512*512 is an optimal image size that allows fast processing speed and accurately represents the characteristics of the learning outcomes. Lastly, we unified the training steps and learning rate in the process of generating the fine-tuned model. To prevent the occurrence of outliers in the learning results, we adopted the optimal values of a learning rate of 2×10^{-6} and 400 training steps, which were derived through multiple experiments conducted by HuggingFace, an institution that utilizes the stable diffusion model and conducts in-depth research on it.

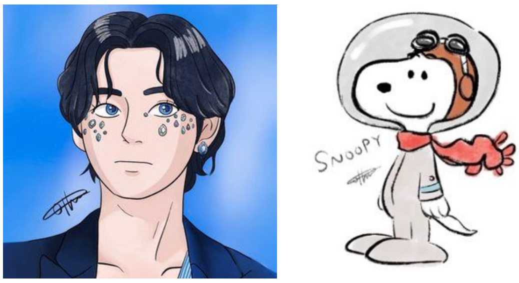


Figure 10 Examples of excluded illustrations

4.1.3 Configuration of Experimental Data

After the Fine-Tuning process, we aimed to generate new images by training the neural network on new illustrations that were not present in the LAION dataset. To achieve this, we obtained drawings from two different illustrators with their research consent and used them for Fine-Tuning. For each illustration by each illustrator, we trained the model with approximately 10 photos and generated keywords specific to each illustrator. Following the Fine-Tuning paper's implementation, we used the returned .ckpt file after the completion of training and applied it to the Stable Diffusion environment to verify if the generated images were appropriate. In order to create a more accurate Fine-Tuning model during the data collection process, we went through several steps. Firstly, we did not consider whether the drawings by the illustrators were for the same character. We included drawings that were not specifically related to the same character in the training data. Secondly, if excessive embellishments hindered the model from recognizing the true form of the character, we excluded such drawings from the data. Thirdly, if the illustrations imitated characters other than the illustrator's original work, we excluded them from the training data. The following are some examples of drawings that were excluded from the training due to these reasons.

For photos like the one on the left in [Figure 10], the decorations drawn below the eyes in the illustration had a negative impact on the model during the Fine-Tuning process. For photos like the one on the right, where the character's drawing imitated another character, we excluded them from the training data due to concerns about the negative effects on learning the illustrator's original drawing style.

In other cases, we chose not to exclude the drawings as excessive subjectivity could be a problem. Here are examples of the drawings used in the training process for each illustrator:

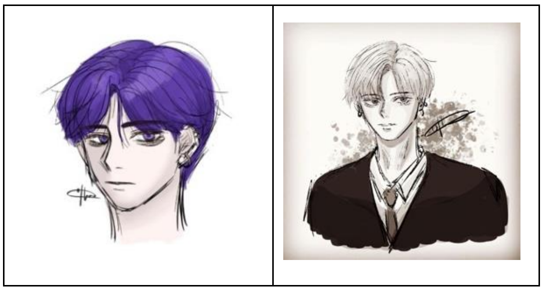


Figure 11 Examples of illustrations by the first illustrator

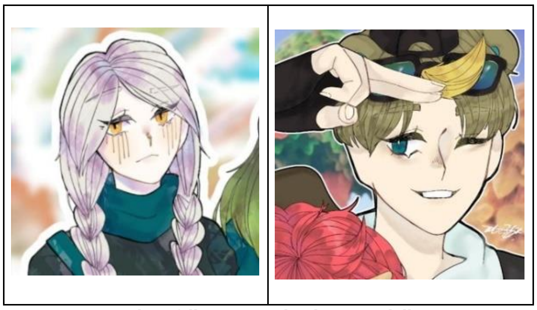


Figure 12 Examples of illustrations by the second illustrator

4.1.4 Experimental Process

Using the Fine-Tuning process described in 4.1.3, we were able to obtain an image generation model that includes new images. Therefore, in this study, we proceeded with the experiment through the following steps to validate our research hypothesis. First, we trained the model using the original image dataset. During this process, we assigned an image keyword, [A], and enabled the generation of images using that keyword. After the Fine-Tuning of the model was completed, we verified if the model successfully learned the style or character forms of the trained images using the assigned keyword. Then, we used the hypotheses formulated earlier and implemented a program to modify the images with parameters. These modified images were also subjected to Fine-Tuning in the same manner. Then, we evaluated the similarity between the training data and the generated image from the modified image. We repeated the same process for the second illustrator.

Through this process, we anticipated that we would be able to objectively verify the decrease in the learning rate of the generation model due to image modifications.

4.1.5 Validation of Output Data

To evaluate the quality of the generated images obtained through the experimental process described in 4.1.4, we introduced the G-FID (Generative Frechet Inception Distance) score. As mentioned before, the G-FID score is utilized as a metric to assess the quality of generated images. In this study, we conducted quantitative quality evaluation using the G-FID score and simultaneously evaluated the performance of the image generation model through qualitative assessment.

In addition to the G-FID score, we also introduced the Inception Score. The Inception Score measures the quality of generated images using entropy, which represents randomness. For instance, when two distributions are given, if one distribution has more uniform values compared to the other distribution, we can say that it has decreased predictability or higher entropy. Therefore, the Inception Score evaluates the diversity of generated images based on how well they can be classified. The more diverse the generated images, the more uniform the distribution returned. [34-35]

The Inception Score can be calculated using the following equation:

$$IS = e^{E_x KL(p(y|x)||p(y))}$$

In this study, we utilized the Inception Score as a metric for evaluating the quality of generated images.

4.1.6 Measurement of Time Taken for Image Modification

Furthermore, we decided it necessary to verify the time required for image modification processing. If the time taken for modification is excessively long, it can be difficult for users to use the images in the process of distribution. Therefore, we aimed to ensure that image modification is completed within a reasonable time frame by measuring the time required for modification.

5. Results

5.1 Experiment on Learning Prevention through Image Modifications

5.1.1 Fine-Tuning with Original Image Data

Using the Fine-Tuned Stable Diffusion image generation model generated through the processes described in 4.1.2 and 4.1.3, various prompts were presented, and the resulting output image data was examined. The results are presented in the order of the first and second illustrators in [Figure 11] and [Figure 12].

Firstly, the following two illustrations are the results of generation without any specific prompts, but with assigned keywords. As mentioned in the Fine-Tuning process, keywords are associated with the image dataset to be used as inputs in the Stable Diffusion phase. For convenience, we set keyword [A] for the images of the first illustrator, and the keyword [B] for the images of the second illustrator. The generated images are as follows.

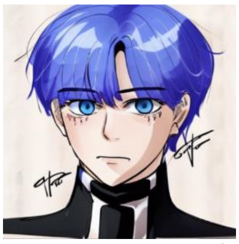


Figure 13 Image generated with prompt [A]

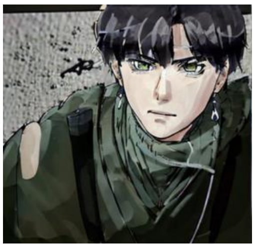


Figure 14 [A] Image generated with prompt [A]

When analyzing the images generated without a separate prompt for the keyword, we can observe that there is no awkwardness in the form or color of the images. The generated illustrations successfully mimic the original style, including the shape of the eye, face structure, texture of the digital paint tool used, colors, and even the unique signature of the illustrator. Next, we provided a prompt that includes additional parameters to the keyword [A]. The following are generated images using additional prompts.

The two images above are generated by providing keywords that can specify the context in addition to the [A] prompt. Similarly, these images successfully mimicked the illustrator's style, with no awkwardly generated parts and a close resemblance to the drawing style in the training data. Moreover, the model was able to successfully generate objects like books and pens that do not exist in the training data, while still matching the illustrator's style. The following are the image generation results from the fine-tuned model with the second illustrator's artwork.

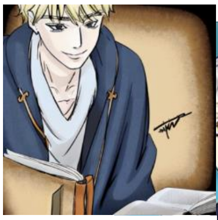


Figure 15 Image generated using prompt ‘[A] reading a book ‘

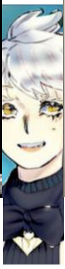


Figure 17 Generated image with prompt [B]

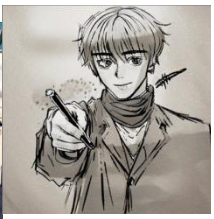


Figure 16 Image generated using prompt '[A] holding a pen'

As you can see, even for objects that do not exist in the training data, the representation is not awkward, and the generated images successfully mimic the original illustrator's style without any awkwardness. The colors and distinctive backgrounds also appear in a similar style. Through these experiments, we confirmed that the Stable Diffusion model can generate new images that match the illustrator's style, even with a simple Fine-Tuning process.

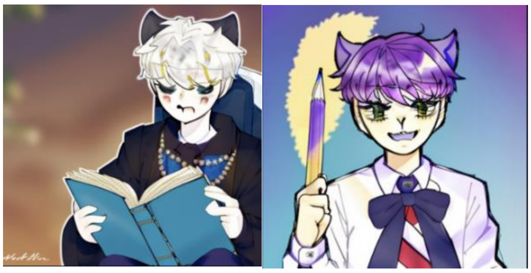


Figure 18 Generated image for [B] with the prompt "reading a book, holding a pen"

5.1.2 Fine-Tuning with Modified Images

Through 5.1.1, we have demonstrated the feasibility of learning the illustration style, which was one of the initial conditions proposed in this study. By simply training on around 10 illustrations without any additional information, we proved that the model learned not only the illustration style but also the texture and signature of the drawings. Next, we experimented with reducing the learning rate through image modifications, as mentioned earlier.

The code used in the experiment is the one provided in Code 1. In this process, the parameters consist of the values for kernelSize and nearbyBlurSize. Following the notation introduced in 3.1.2, we set the transformations to (5, 3) and (3, 3) respectively and applied them to the images.

Firstly, when we applied the modifications to the example images shown in 4.1.3, the results were as follows:

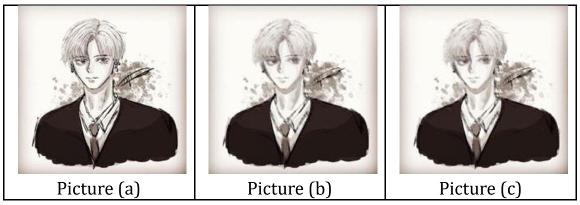

Picture (a)
Picture (b)
Picture (c)

Figure 19 Difference in image due to modification parameters

In the three images above, Picture (a) is the original image, Picture (b) is the image transformed with parameters (5, 3), and Picture (c) is the image transformed with parameters (3, 3). During the training process of 5.1.1, all the images used were transformed with both (5, 3) and (3, 3) parameters, and they were separately used in the new Fine-Tuning process. This resulted in a total of four Fine-Tuning models, which were then used to generate images using the approach described in 5.1.1. First, for A, the generated image results using the Fine-Tuning model after transformation are as follows.

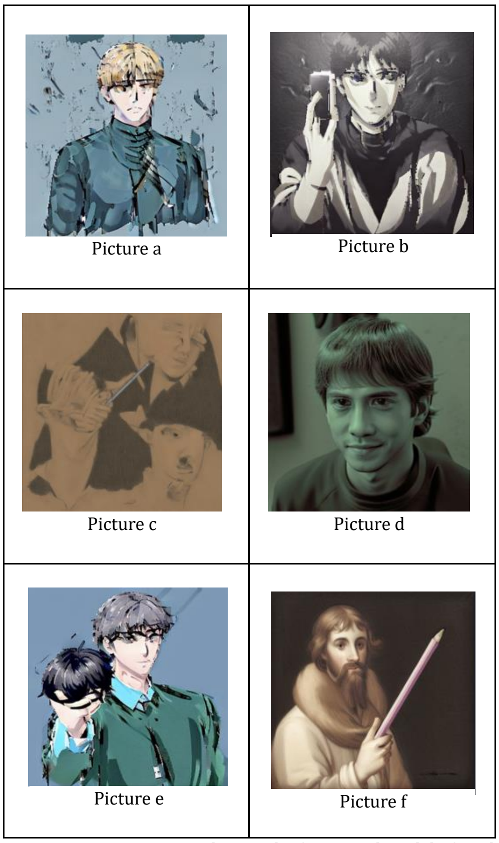

Picture a
Picture b
Picture c
Picture d
Picture e
Picture f

Figure 20 Image generated using the fine-tuned model of modified A.

Among the six images above, Pictures a~c are the results of models trained on images modified with parameters (5, 3), and Pictures d~f are the results of models trained on images modified with parameters (3, 3). Picture a and d were generated without any specific prompts, b and e were generated with the prompt '[A] holding a book', and c and f were generated with the prompt '[A] holding a pen'. When qualitatively analyzing these six images, it can be concluded that the reduction in learning rate through image modifications was successfully demonstrated.

Firstly, in the case of Pictures a, b, and e, as the model learned the elements within the blurred areas of the transformed images, the generated images exhibit significant awkwardness and distortion in features such as eyes, chin, and hair. Moreover, in the case of Picture e, we speculated that there was a misinterpretation within the model where the hair element was mistaken for the book element, resulting in the presence of hair instead of a book. Picture c clearly indicates more errors in the generation process. The shape of the face is unrecognizable as [A], and even within the face, the eyes, nose, and mouth are not clearly discernible, with an abnormal arrangement. Although the prompt included the word "pen," it is difficult to determine the presence of a normal body and face. In Pictures d and f, due to the larger modification parameters (3, 3), the original image's art style is completely unrecognizable, and instead, highly realistic photos are generated, making it challenging to determine whether the Fine-Tuning process was conducted normally. The images generated from the models Fine-Tuned on modified images not only exhibited some low-quality aspects but failed to generate any normal images. This phenomenon was also observed in the case of [B] illustrations, where the same process was applied for Fine-Tuning, and the generated images are as follows. However, due to space constraints, we included only three example images that closely resemble the generated results for [A] in the paper.

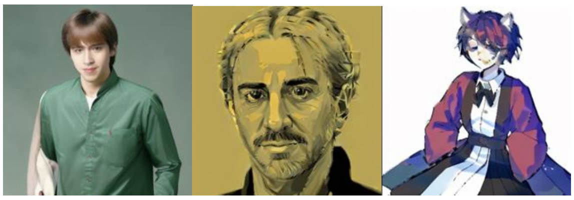


Figure 21 Image generated using the fine-tuned model of modified B.

As can be observed from the three images above, the art style of the B illustration, as presented in 4.1.3, has been lost. In the case of the image on the right, it has the closest resemblance to the original image among the generated images. However, the facial features have significantly deteriorated, with several distorted and disconnected parts such as eyes, nose, and mouth.

Through these experimental processes, we have qualitatively validated the hypothesis of reduced learning efficiency through image modifications, as stated in the early stages of the paper.

Therefore, we aimed to verify whether users would have difficulty recognizing such images when they are distributed online after undergoing transformations.

5.1.3 Quantitative Analyzation of Generated Images

In 5.1.2, we conducted a qualitative analysis of the image quality generated by the Fine-Tuned model using the modified image dataset. Although this process has demonstrated a significant decrease in the generated image quality, we aim to provide more objective evidence through quantitative metrics. To achieve this, we utilized the G-FID (Generative Frechet Inception Distance) and Inception Score.

First, we performed Fine-Tuning using the (5, 3) transformed datasets of A and B illustrations. Then, we generated 50 images using the trained model in the Stable Diffusion framework without providing any prompts. Subsequently, we calculated the Inception Score and G-FID scores for these 50 images, computing the mean, standard deviation, and checking for outliers. However, as mentioned before, we excluded images that significantly deviate from the learned art style, such as Picture d in Table 2 of section 5.1.2, considering them as outliers.

The code for calculating the Inception Score and G-FID scores is as follows.

```
(…)
def calculate_inception_score(
   sample_dataloader,
   test_dataloader,
   device="cpu",
   num_images=50000,
   splits=10,
):
   inception_model = InceptionScore(device=device)
   inception_model.eval()
(…)
   for k in range(splits):
      part = preds[k * (num_images // splits) : (k + 1) *
(num_images // splits), :
      ]
      py = np.mean(part, axis=0)
      scores = []
      for i in range(part.shape[0]):
         pyx = part[i, :]
         scores.append(entropy(pyx, py))
      split_scores.append(np.exp(np.mean(scores)))

   return {"Inception Score": np.mean(split_scores)}
```

Code 2 Function for calculating Inception Score

```
(…)
def calculate_gfid(mu1, sigma1, mu2, sigma2, eps=2e-6):
(…)
   covmean, _ = linalg.sqrtm(sigma1.dot(sigma2), disp=False)
   if not np.isfinite(covmean).all():
      msg = ('fid calculation produces singular product; '
            'adding %s to diagonal of cov estimates') % eps
      print(msg)
      offset = np.eye(sigma1.shape[0]) * eps
      covmean = linalg.sqrtm((sigma1 + offset).dot(sigma2 +
offset))

   if np.iscomplexobj(covmean):
      if not np.allclose(np.diagonal(covmean).imag, 0,
atol=1e-3):
         m = np.max(np.abs(covmean.imag))
(…)
      covmean = covmean.real

   tr_covmean = np.trace(covmean)

   return (diff.dot(diff) + np.trace(sigma1)
         + np.trace(sigma2) - 2 * tr_covmean)
```

Code 3 Function to calculate G-FID

The statistical values of Inception Score and G-FID calculated using the two types of Python code mentioned above for A and B are as follows. The significant figures are set to 5 decimal places for Inception Score and 1 decimal place for G-FID.

	Inception Score	G-FID
Average	0.00926	275.8
Standard Deviation	0.00063	44.0

Table 2 Image Quality Assessment on image dataset A

	Inception Score	G-FID
Average	0.01526	249.2
Standard Deviation	0.00053	28.4

Table 3 Image Quality Assessment on image dataset B

The experimental results showed extremely low values, which are consistent with the qualitative evaluation conducted earlier. Referring to the FID and Inception Score provided and the designed experiment, the given values indicate a very high disturbance in the generated model, leading to the loss of functionality as an image generation model. Through this process, we were able to quantitatively demonstrate the impact of learning rate degradation through image modifications.

5.1.4 User Recognition of Modified Images

Through the experiment, it was observed that when images were modified, both the Stable Diffusion model and the Fine-Tuning process experienced increased inaccuracies, making it impossible to generate the intended images as perceived by the users. As a result, this study proposed the modification method to prevent unauthorized learning by Text-to-Image AI when illustrators or cartoonists distribute their own works. However, in order to achieve this, the modified images should provide maximum possible inaccuracy to the generation model while not significantly hindering general users' recognition of the images. Therefore, in Section 5.1.3, we aimed to assess the extent of quality degradation of the modified images compared to the original images and verify that users have no difficulty in recognizing the images.

To set up the experimental conditions, the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) were introduced as metrics to evaluate the quality of the modified images.

Firstly, using a dataset of 11 images modified with a distortion level of (5, 3) from A, the PSNR and SSIM values were calculated compared to the original photos. The calculated results are as follows. The significant figures were set to three decimal places, and no outliers beyond $Q1 - 1.5 \times IQR$ or $Q3 + 1.5 \times IQR$, that is, $\mu \pm 2.7\sigma$, were found in the experimental results.

	Average	Standard Deviation
PSNR (dB)	27.867	8.27
SSIM	0.923	0.051

Table 4 Quality of image dataset A modified with distortion level (5, 3)

Considering that the calculated values are above an average PSNR value of 25 and an SSIM value of 0.9, it can be inferred that the image distortion is not severe, and significant issues in user recognition of the images are unlikely to occur. Similarly, for B, 13 images were modified with a distortion level of (5, 3), and the quality metrics were computed and statistically summarized. No outliers were found for B as well.

	Average	Standard Deviation
PSNR (dB)	26.981	7.44
SSIM	0.927	0.054

Table 5: Quality of B image set modified with distortion level (5, 3)

Overall, through the calculation of PSNR and SSIM values between the modified images and the original images for A and B, it was determined that image modification has a significant impact on the performance degradation of the image generation model. However, considering the acceptable range of image quality as suggested in previous studies, we concluded that it would not significantly impede user recognition. Thus, we confirmed that the reduction in training efficiency of the image generation model through image distortion is effective and does not impose significant obstacles for general users in viewing images. Additionally, based on the previous studies, the mechanism of human perception in recognizing images and objects was taken into account to minimize interference in human image recognition. This ensures prevention of unauthorized image learning while minimizing discomfort for users when viewing images.

5.1.5 Time Measurement for Image Modification Process

Additionally, the time required for image modification was measured. This was to ensure users do not face difficulties when applying the modification method proposed in this study to prevent unauthorized image training, especially if the modification process takes excessive time.

The time required to apply the transformation effect to a total of 24 images included in the A and B image datasets was measured. The calculation revealed an average time of approximately 1.8 seconds. However, there were values that exceeded the range of +2.7 sigma in the experimental results. These values were attributed to the higher number of detected edges during the Canny Edge Detection process compared to other images, resulting in an increase in the number of pixels that required blur effects. Ultimately, we determined that the 1.8-second modification time would not cause significant inconvenience to users, and users intending to proceed with the modification process would be able to do so comfortably.

6. Conclusion

In this study, it was confirmed that indiscriminate data collection occurs during the training process of Text to Image AI, represented by Stable Diffusion. In this process, unauthorized works are used for training, posing a problem where copyright holders face difficulties in protection. To address this issue and prevent the training of one's own works by third-party artificial intelligence, a solution was proposed. The solution involves slightly modifying the images in a way that does not significantly hinder users' image perception, thereby disrupting the training process of the AI. To validate these hypotheses, a program for image modification was implemented, and the modified images were trained on the DreamBooth Fine-Tuning model to generate images. The results showed that the model trained on the original illustrations successfully captured the style, texture, and artistry of the illustrations, while the model trained on slightly modified illustrations exhibited a significant decrease in training effectiveness, generating images of very low quality. These findings were confirmed not only through qualitative evaluation but also through the introduction of G-FID scores and Inception Scores. Furthermore, the convenience of viewing modified images by general users was assessed using SSIM and PSNR scores. Therefore, it is anticipated that the methodology proposed in this study will contribute to preventing unauthorized learning of copyrighted works and ensuring rights protection for copyright holders.

References

[1] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[2] Stability AI. Official documentation. Cited Nov 14, 2022. Available from: https://stability.ai/blog/stable-diffusion-public-release. (accessed Nov 14, 2022).

[3] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." Communications of the ACM 63, no. 11 (2020): 139-144.

[4] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

[5] Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

[6] Brock, Andrew, Jeff Donahue, and Karen Simonyan. "Large scale GAN training for high fidelity natural image synthesis." *arXiv preprint arXiv:1809.11096* (2018).

[7] Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

[8] Ramesh, Aditya, et al. "Zero-shot text-to-image generation." *International Conference on Machine Learning*. PMLR, 2021.

[9] Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684-10695. 2022.

[10] Luisier, Florian, Thierry Blu, and Michael Unser. "Image denoising in mixed Poisson–Gaussian noise." IEEE Transactions on image processing 20.3 (2010): 696-708.

[11] https://laion.ai/blog/laion-5b/ LAION AI. Official blog post. Cited Nov 28, 2022. Available from: https://laion.ai/blog/laion-5b (accessed Nov 28, 2022).

[12] Ruiz, Nataniel, et al. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation." arXiv preprint arXiv:2208.12242 (2022).

[13] Lawrence W. Stark, Claudio M. Privitera, Huiyang Yang, Michela Azzariti, Yeuk Fai Ho, Theodore T. Blackmon, and Dimitri A. Chernyak "Representation of human vision in the brain: how does human perception recognize images?," Journal of Electronic Imaging 10(1), (1 January 2001). https://doi.org/10.1117/1.1329895

[14] Kanwisher, Nancy. "Functional imaging of human visual recognition Nancy Kanwisher"", Marvin M. Chun", Josh McDermott", Patrick J. Ledden." *Cognitive Brain Research* 5 (1996): 55-67.

[15] Haxby, James V., Leslie G. Ungerleider, Barry Horwitz, Jose Ma Maisog, Stanley I. Rapoport, and Cheryl L. Grady. "Face encoding and recognition in the human brain." *Proceedings of the National Academy of Sciences* 93, no. 2 (1996): 922-927.

[16] Wardle, Susan G., and Chris I. Baker. "Recent advances in understanding object recognition in the human brain: deep neural networks, temporal dynamics, and context." *F1000Research* 9 (2020).

[17] Raichle, Marcus E. "A brief history of human brain mapping." *Trends in neurosciences* 32, no. 2 (2009): 118-126.

[18] Ding, Lijun, and Ardeshir Goshtasby. "On the Canny edge detector." *Pattern recognition* 34, no. 3 (2001): 721-725.

[19] Xu, Zhao, Xu Baojie, and Wu Guoxin. "Canny edge detection based on Open CV." In *2017 13th IEEE international conference on electronic measurement & instruments (ICEMI)*, pp. 53-56. IEEE, 2017.

[20] Huynh-Thu, Quan, and Mohammed Ghanbari. "Scope of validity of PSNR in image/video quality assessment." *Electronics letters* 44, no. 13 (2008): 800-801.

[21] Hore, Alain, and Djemel Ziou. "Image quality metrics: PSNR vs. SSIM." In *2010 20th international conference on pattern recognition*, pp. 2366-2369. IEEE, 2010.

[22] Sara, Umme, Morium Akter, and Mohammad Shorif Uddin. "Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study." *Journal of Computer and*

Communications 7, no. 3 (2019): 8-18.

[23] Channappayya, Sumohana S., Alan Conrad Bovik, and Robert W. Heath. "Rate bounds on SSIM index of quantized images." *IEEE Transactions on Image Processing* 17, no. 9 (2008): 1624-1639.

[24] Han, Yu, Yunze Cai, Yin Cao, and Xiaoming Xu. "A new image fusion performance metric based on visual information fidelity." *Information fusion* 14, no. 2 (2013): 127-135.

[25] Naeem, Muhammad Ferjad, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. "Reliable fidelity and diversity metrics for generative models." In *International Conference on Machine Learning*, pp. 7176-7185. PMLR, 2020.

[26] Moffat, Alistair. "Huffman coding." *ACM Computing Surveys (CSUR)* 52, no. 4 (2019): 1-35.

[27] Knuth, Donald E. "Dynamic huffman coding." Journal of algorithms 6, no. 2 (1985): 163-180.

[28] Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." *arXiv preprint arXiv:1510.00149* (2015).

[29] Kreft, Sebastian, and Gonzalo Navarro. "LZ77-like compression with fast random access." In 2010 Data Compression Conference, pp. 239-248. IEEE, 2010.

[30] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." (1999).

[31] Bogdanov, Andrey, Dmitry Khovratovich, and Christian Rechberger. "Biclique cryptanalysis of the full AES." In *International conference on the theory and application of cryptology and information security*, pp. 344-371. Springer, Berlin, Heidelberg, 2011.

[32] Sharma, Dushyant, Rishabh Shukla, Anil Kumar Giri, and Sumit Kumar. "A brief review on search engine optimization." In *2019 9th international conference on cloud computing, data science & engineering (confluence)*, pp. 687-692. IEEE, 2019.

[33] Gudivada, Venkat N., Dhana Rao, and Jordan Paris. "Understanding search-engine optimization." Computer 48, no. 10 (2015): 43-52.

[34] Barratt, Shane, and Rishi Sharma. "A note on the inception score." *arXiv preprint arXiv:1801.01973* (2018).

[35] Obukhov, Artem, and Mikhail Krasnyanskiy. "Quality assessment method for GAN based on modified metrics inception score and Fréchet inception distance." In *Proceedings of the Computational Methods in Systems and Software*, pp. 102-114. Springer, Cham, 2020.