electronics

MDPI

Article

Personalized Text-to-Image Model Enhancement Strategies: SOD Preprocessing and CNN Local Feature Integration

Mujung Kim [1], Jisang Yoo [1,*] and Soonchul Kwon [2]

[1] Department of Electronic Engineering, Kwangwoon University, Seoul 01897, Republic of Korea; kmj1026@kw.ac.kr
[2] Graduate School of Smart Convergence, Kwangwoon University, Seoul 01897, Republic of Korea; ksc0226@kw.ac.kr
* Correspondence: jsyoo@kw.ac.kr; Tel.: +82-2-940-8637

Abstract: Recent advancements in text-to-image models have been substantial, generating new images based on personalized datasets. However, even within a single category, such as furniture, where the structures vary and the patterns are not uniform, the ability of the generated images to preserve the detailed information of the input images remains unsatisfactory. This study introduces a novel method to enhance the quality of the results produced by text-image models. The method utilizes mask preprocessing with an image pyramid-based salient object detection model, incorporates visual information into input prompts using concept image embeddings and a CNN local feature extractor, and includes a filtering process based on similarity measures. When using this approach, we observed both visual and quantitative improvements in CLIP text alignment and DINO metrics, suggesting that the generated images more closely follow the text prompts and more accurately reflect the input image's details. The significance of this research lies in addressing one of the prevailing challenges in the field of personalized image generation: enhancing the capability to consistently and accurately represent the detailed characteristics of input images in the output. This method enables more realistic visualizations through textual prompts enhanced with visual information, additional local features, and unnecessary area removal using a SOD mask; it can also be beneficial in fields that prioritize the accuracy of visual data.

Keywords: diffusion network; image similarity comparison; personalized image generation; salient object detection; text-to-i mage

Citation: Kim, M.; Yoo, J.; Kwon, S. Personalized Text-to-Image Model Enhancement Strategies: SOD Preprocessing and CNN Local Feature Integration. *Electronics* **2023**, *12*, 4707. https://doi.org/10.3390/electronics12224707

Academic Editor: Maciej Ławryńczuk

Received: 26 October 2023
Revised: 13 November 2023
Accepted: 17 November 2023
Published: 19 November 2023

1. Introduction

Recent advancements in text-to-image models [1–6] have considerably improved the quality of generated images. Models like Stable Diffusion [1], which rely on large-scale datasets, can create realistic and imaginative images by capturing the intricate relationship between images and text grounded in extensive image-text pair data. Consequently, users can now perform various application tasks, such as stylization [7–10] and editing [11–13]. However, users' ability to create images aligned with individual conceptualizations is limited. The results often fall short of expectations, even when users provide prompts with specific descriptions tailored to individual concepts, because the vast image-text pair data used for training lack information about personal concepts. Such a problem becomes an obstacle in applications where the precise image generation of the desired object is necessary. In order to address this, recent research in personalized text-to-image models [14–18] has involved learning additional concepts from user image sets. Textual inversion [16] inverts the input image into the text embedding space and subsequently learns new pseudo-words. DreamBooth [14] fine-tuning diffusion models use several user-provided images and unique identifiers. Custom Diffusion [15] enhances memory efficiency by fine-tuning learning parameters in specific layers, enabling users to create personalized

images based on newly learned words from their prompts. Figure 1 illustrates examples of samples generated using these models and our method.

Figure 1. Images generated using the personalized text-to-image models. We generate a personalized image using the input image representing the personal concept and the text prompt. We fine-tune the text-to-image model to ensure that the identifier <V> embedded in the prompt can encapsulate information about the concept. Our approach can preserve concept details by directly infusing visual information into the identifier.

Nevertheless, Despite these advancements, certain issues, as exemplified in Figure 2, persist, especially in categories like furniture, where structures and shapes vary significantly within the same class and where attention to detail is essential for high fidelity. Additionally, as illustrated in Figure 3, when input images learn not only the desirable concept of the object but also unintended concepts (such as the background), the quality of the image degrades. This leads to a mix of undesirable elements in the generated images, which lowers the fidelity of the images or hinders the creation of images that align with the text prompts, thus decreasing text-image alignment. This is especially detrimental in the creation of images for catalogs or advertisements, where an accurate depiction of the product's condition is critical.

Figure 2. Samples that failed to preserve the concept of the input images. The generated samples in the **second row** fail to maintain the details in the input images (**first row**). Notable changes in color, shape, and pattern can be observed.

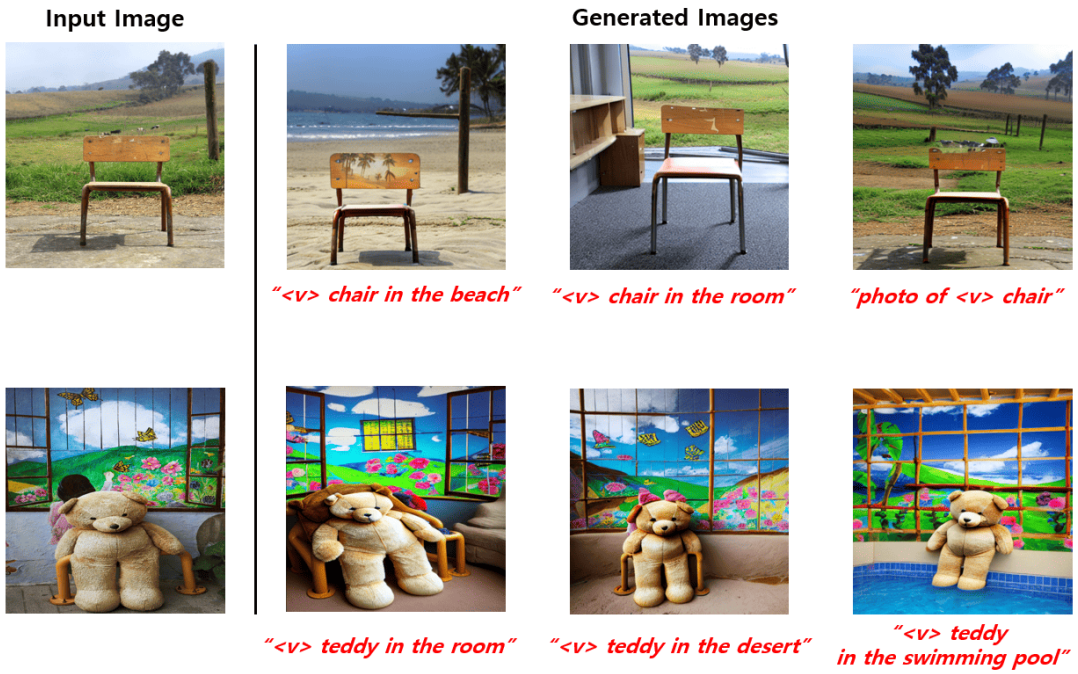

Figure 3. Poorly created image due to unnecessary concepts. Due to the background information included in the image other than the target object, the details of the object were stained, or the image was created in a way that did not fit the situation required by the prompt.

This study introduces several novel methods that incorporate careful steps to address the challenges of the above-mentioned detail preservation and quality degradation. The study conducted experiments utilizing the Custom Diffusion [15] fine-tuning technique. Custom Diffusion fine-tunes Stable Diffusion [1] that was pretrained on large-scale text-image paired data. During this fine-tuning process, Custom Diffusion updates the parameters of specific layers only, thereby enhancing memory efficiency and processing speed. Based on this Custom Diffusion, the new strategies we have introduced are as follows. First, to prevent nontarget features from influencing the generated results, superfluous background information is removed during preprocessing. This ensures a representation that is focused solely on the specific object of interest, eliminating potential distractions or interferences from the background. In order to achieve this, a mask for the desired object within the concept image is extracted using InSPyReNet [19], a salient object detection (SOD) model. InSPyReNet was chosen for its superior capability in salient object detection, especially for high-resolution images. Subsequently, the concept image data, with the extraneous background removed via the extracted mask, is augmented and employed as the training dataset. Second, the concept image is mapped into the textual word embedding space for further learning. The method maintains a detailed representation of the object by feeding information about the concept image into the text prompt, which acts as a condition in the diffusion network. Unlike previous methods that relied on text-image attention mechanisms, our approach utilizes a pretrained CLIP image encoder to extract image embeddings from concept images. Additionally, we introduced a CNN network, ResNet-50 [20], to extract local features from these images, providing additional information on shapes, forms, and patterns. By extracting local features from the intermediate layers of a pretrained ResNet and combining them with image embeddings, we achieved a comprehensive feature representation. This combined feature information was then injected into the text embeddings corresponding to the identifier prompts. This method preserves details and structural information better than previous methods that initialized identifiers with arbitrary words. Third, using a Siamese network [21], the similarity between the images generated in the postprocessing step and the training images was assessed, and the results that fell below a predetermined threshold were discarded. Finally, quantitative and

qualitative evaluations were conducted, comparing our approach with existing models. By applying our strategy, we observed improved results in terms of detail retention performance, as evidenced by the increased text alignment scores and DINO image alignment scores. The comprehensive workflow is illustrated in Figure 4.

We recognize several concurrent studies with similar themes [17,18]. ELITE [17] develops a training network that maps visual inputs to multiple textual embeddings using multi-layer embeddings, fine-tuning the attention layers of a pretrained text-image model by projecting the foreground object into the textual feature space. On the other hand, Instanbooth [18] utilizes a learnable image encoder to convert input images into textual tokens, employing these as conditions for the cross-attention layers. It also learns visual features through separate adaptor layers and encoders for fine details. Our approach, which is similar to ELITE and Instanbooth, employs image encoders to map acquired image tokens into the textual space. However, unlike these methods, we introduce a separate network for extracting local features, combining these extracted features with the tokens to create new embeddings. These new embeddings are then used to fine-tune the attention layers. Additionally, we employ cosine similarity between the embeddings of the generated sample images and concept images during the training process, guiding the generated samples to more closely resemble the input images.

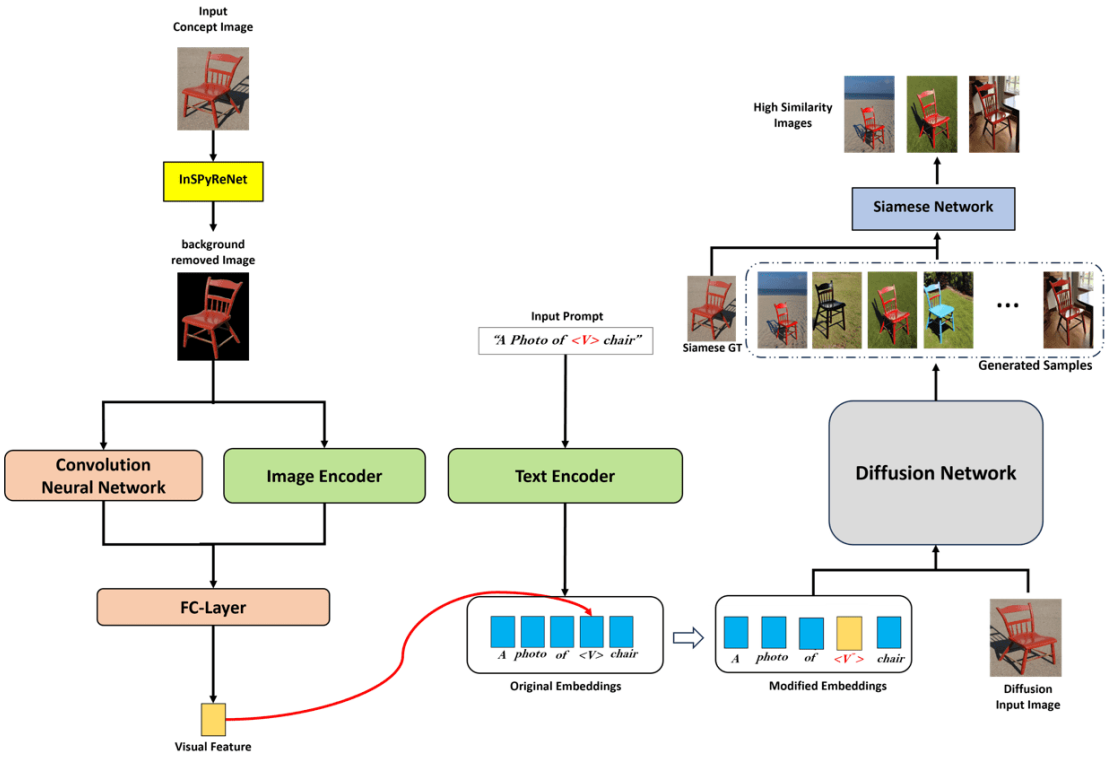

Figure 4. Pipeline of our proposed method. (1) From the concept image, we remove the background using the SOD model, InSPyReNet [19], and then obtain image embeddings using the CLIP [22] image encoder. (2) We extract local features from the concept image using CNN. (3) We concatenate the image embeddings with the extracted local features to form a new visual feature. Next, we replace the embeddings corresponding to the identifier in the original prompt using the text encoder to obtain the modified embeddings.(4) The Siamese network [21] is then used to measure the similarity between the samples generated through the modified prompt embeddings and concept images. Finally, we obtain images with high similarity as the final results.

In summary, the contributions of our study are as follows:

- We have introduced the use of a salient object detection (SOD) mask in the preprocessing phase to remove information other than the prominent object. This ensures

that the image generation process is focused on the target object, thereby avoiding the degradation of image quality by irrelevant information.
- Instead of relying on text embeddings that represent identifiers, we have mapped the image embeddings obtained from concept images and provided additional local features. This has improved the detail preservation performance of our models.
- We have employed a Siamese network in the postprocessing phase to compare the similarity between the generated images and the concept images, which allows for quality control. This ensures that only the images with high fidelity are selected, enhancing the overall quality of the output.

2. Related Work

2.1. Text-to-Image Models

The text-to-image model generates images based on user-provided text, allowing users to influence the resulting image directly. Recently, deep learning-based text-to-image models have garnered significant attention. Current research in deep text-to-image models primarily centers around generative adversarial networks (GANs) [23–27], variational autoencoders (VAEs) [28,29], and diffusion-based models [1,2,14]. However, GAN and VAE-based models exhibit limitations, particularly when precise objects or feature placements are required. Moreover, These models struggle when generating images with intricate patterns and structures, such as faces, eyes, noses, mouths, or complex decorations. Furthermore, even when these models produce reasonably plausible images, they fall short of closely aligning with the provided text prompts. In contrast, diffusion-based models leverage extensive training datasets containing text-image pairs to generate more realistic and intricate images. Examples include DALL-E [30], which has demonstrated impressive results by employing an autoregressive model. DALL-E2 [2], Imagen [3], Stable Diffusion [1], and others incorporate large-scale text encoders based on data, enabling enhanced control during image synthesis. Moreover, researchers are increasingly harnessing the control capabilities offered by pretrained diffusion-based models with extensive image-text data for image editing and style transfer. SINE [31] employs a pretrained large-scale diffusion model for single-image editing and style transfer. Additionally, GLIGEN [32] explores image inpainting by introducing additional layers and incorporating various conditions beyond text, such as bounding boxes and keypoints. ControlNet [33] is a neural network designed to add spatial condition control to large-scale pretrained text-to-image diffusion models. It safely adjusts those parameters that leverage "zero convolution" and demonstrates robust learning under various conditions and across large and small datasets.

2.2. Personalized Image Generation

While text-to-image models [31–33] have made significant strides in providing precise control with textual guidance, their generated images are often limited to generating general instances. In contrast, personalized image generation takes user-defined concepts as input, allowing for the precise editing and transformation of these concepts. Numerous studies [14–18] have delved into this domain, employing various techniques to achieve personalized image manipulation. In GAN-based models, the GAN-inversion method [10,26,27,34–36] has been commonly employed for image editing and personalized image creation. The method projects an image directly into the latent space, obtains an edited latent code, and subsequently generates the edited image through the generator process. GAN-based approaches have primarily been used for tasks like overall image style transfer [10,26,27], facial expression changes [35,36], and age modifications [34]. More recently, methods for personalized image generation have emerged, leveraging pretrained large-scale text-to-image models. This approach, known as Textual Inversion [16], discovers new embeddings within the embedding space that represent the user-provided visual concept. Subsequently, a new image is generated using the pseudo-word associated with this embedding. Similar to text inversion, DreamBooth [14] takes an image representing the concept as input and uses the information corresponding to the instance's class as input; it then fine-tunes it and encodes it into a unique identifier.

This method allows for learning new concepts with higher fidelity and addresses language drift. Custom diffusion [15], an extension of this technique, has demonstrated satisfactory performance improvements with faster fine-tuning, achieved by updating only the parameters of the cross-attention layer. ELITE [17] employs local mapping and multi-layer global mapping networks to preserve the details when encoding visual concepts into textual embeddings. Similarly, InstantBooth [18] maps input images to the textual space and introduces adapter layers to inject identity information from the input images into the backbone model.

2.3. Salient Object Detetion

SOD aims to identify and segment the most attention-grabbing object or region within an image. Hou et al. [37] incorporated short connections into the skip-layer structure of the holistically nested edge detection [38] framework. Each layer within this architecture yields rich multi-scale feature maps. Xie et al. [39] addressed the challenges posed by the shallow layers of the backbone network, which struggle to acquire global semantic information. Incorporating fully convolutional networks [40] and multi-path recurrent feedback mechanisms was instrumental in enhancing performance. Moreover, Pang et al. [41] proposed aggregate interaction modules to effectively integrate features from neighboring levels while mitigating noise. Additionally, the InSPyReNet [19] framework introduced a novel pyramid blending method, which systematically synthesizes two distinct pyramids derived from low- and high-resolution scales for high-resolution SOD.

2.4. Image Similarity Comparison

Traditional methods for comparing image similarity encompass pixel-based [42] and structural feature-based approaches [43–45]. Pixel-based methods that rely on direct pixel comparisons are susceptible to variations in lighting, scale, or viewing angles. Therefore, the structural similarity index [42] was introduced to capture perceptual changes in images rather than mere pixel-level differences. In contrast, feature-based methods, such as scale-invariant feature transform [43] and speeded-up robust features [44], employ key points and descriptors for image comparison. These methods exhibit robustness when dealing with transformations and occlusions. In recent years, the field has witnessed the emergence of neural network-based methods [21,46] for image similarity comparison. The approach involves extracting feature maps from the intermediate layers of pretrained deep learning models, such as VGG [47] or ResNet [20]. Subsequently, metrics such as cosine similarity or Euclidean distance are computed, and if they surpass a predefined threshold, the images are deemed to represent the same object. Another neural network-based approach is the Siamese network [21], comprising two subnetworks that share identical weights. It calculates the similarity distance between feature vectors extracted from two input images. Expanding on this, the triplet network [46] processes three input images, referred to as the anchor, positive, and negative samples. The aim is to ensure that the anchor image is closer in feature space to the positive image (same class) than to the negative image (different class). This study employs the Siamese network to determine whether a generated sample and a reference concept image depict the same object.

3. Method

We aim to generate images that faithfully represent the underlying concept by employing a pretrained text-to-image model [1]. First, we utilized a SOD network [19] to extract masks corresponding to the target salient objects while eliminating extraneous background information unrelated to the concept. In order to integrate visual features from concept images, we concatenated the embeddings derived from the CLIP [22] image encoder with the local features of the concept image extracted using the CNN and replaced segments of the textual prompt embeddings with this integrated information. Next, we evaluated the similarity between the generated samples derived from the adapted prompts and the reference concept images using a Siamese network. Finally, we filtered out any results below a predefined similarity threshold to obtain the final outcome. This chapter presents

an overview of the proposed large-scale text-to-image model, beginning with a discussion of the background information in Section 3.1. Section 3.2 outlines our proposed data preprocessing methods, Section 3.3 details the procedure for inserting image embeddings into text embedding, and finally, Section 3.4 elaborates on postprocessing techniques that leverage image similarity measurements.

3.1. Text-to-Image Diffusion Models

We employ Stable Diffusion [1], a text-to-image diffusion model comprising various components and modules and trained on large-scale image-text pairs. Initially, the autoencoder's encoder (denoted as ε) is trained to map the input image (x) to the spatial latent code ($z = \varepsilon(x)$). The decoder, D, learns to map the latent code back to the image $D(\varepsilon(x)) \approx x$. Moreover, the diffusion model, like other generative models, models the conditional distribution as $p(z|y)$. In the text-to-image task, image generation is controlled based on the input $y(textcondition)$. In order to preprocess y, the CLIP text encoder c_θ sends y to an intermediate representation, and a cross-attention layer calculates the correlation between the text and image. The objective of this conditional latent diffusion model is as follows:

$$L_{LDM} := \mathbb{E}_{z\sim\varepsilon(x),y,\epsilon\sim\mathcal{N}(0,1),t}[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2], \quad (1)$$

where ϵ represents an unscaled noise sample, t denotes the time step, z_t corresponds to the latent noise at time t, and ϵ_θ denotes the denoising network. During inference, random noise is sampled and iteratively denoised to produce a new latent image, z_t. Subsequently, this latent code is transformed into a new image through a pretrained decoder, $x' = D(z_t)$. Inside the model, the cross-attention layer modifies the intermediate representation, $\phi(z_t)$, based on the condition $c_\theta(y)$. First, according to the projection layer, we represent the query as $Q = W_Q \cdot \phi(z_t)$, the key as $K = W_K \cdot c_\theta(y)$, and the value as $V = W_V \cdot c_\theta(y)$. W_Q, W_K, and W_V are the weight parameters of the query, key, and value projection layers, respectively. The attention mechanism is then executed as a weighted sum over the value features.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d'}}\right)V, \quad (2)$$

where d' represents the output dimensions of the key and query. The latent image features are updated using the attention block output. During fine-tuning, we adjust the distribution mapping from images to text, drawing inspiration from Custom Diffusion and specifically updating the parameters W_K and W_V of the diffusion model.

3.2. Data Preprocessing with SOD

If the object for personalization is not clearly highlighted in the concept images used as input data, there can be issues. For example, if multiple objects are captured or if the background has too much influence (as seen in Figure 3), unwanted objects may contribute to the generation process and impede accurate creation. Moreover, there can be issues where backgrounds that do not match the context of the input text are generated as the background of the created image. In order to address these issues, we propose a preprocessing method for concept images. The aim of our proposed method is to detect the object to be preserved in the concept image and filter out all other parts. Therefore, we propose using a salient object detection model that identifies the most visually salient object within the image, as the identification of objects other than the target object is unnecessary. In this study, we used high-resolution images collected from Unsplash and the high-resolution image dataset BIG as our experimental data. To this end, we employed InSPyReNet [19], which has demonstrated superior salient object detection capabilities in high-resolution images.

By using an original concept image set, $\mathbf{X}_o$, containing N images, i.e., $\mathbf{X}_o = \{x_o^1, x_o^2, x_o^3, \ldots, x_o^N\}$, we used InspyRenet to detect only the most salient object in each concept image and obtain the mask m_o^n for this object. m_o^n is the mask for the salient object of the nth concept image, with the object having a pixel value of 1 and the remaining having

a value of 0. By using the concept images and masks for the salient objects, we obtained the final dataset for training, $\mathbf{X_m}$, as follows:

$$x_m^n = x_o^n \cdot m_o^n, \quad n \in \{1, 2, 3, ..., N\} \tag{3}$$

We used X_m with random augmentation during training, and we did not augment it during inference.

Figure 5 illustrates the generation results of our method (using SOD preprocessing) compared to those of previous studies [14–16] that did not remove areas interfering with the preservation of the concept. The prior methods tend to reflect all features included in the concept image, regardless of the content of the prompt. That is, they attempt to replicate not only the bicycle, which is the object to be preserved, but also the blue wall, the beige floor, and even the composition of the wall and floor. In contrast, by applying our preprocessing method, we can see that the identity of the target object is preserved without being affected by unnecessary regions.

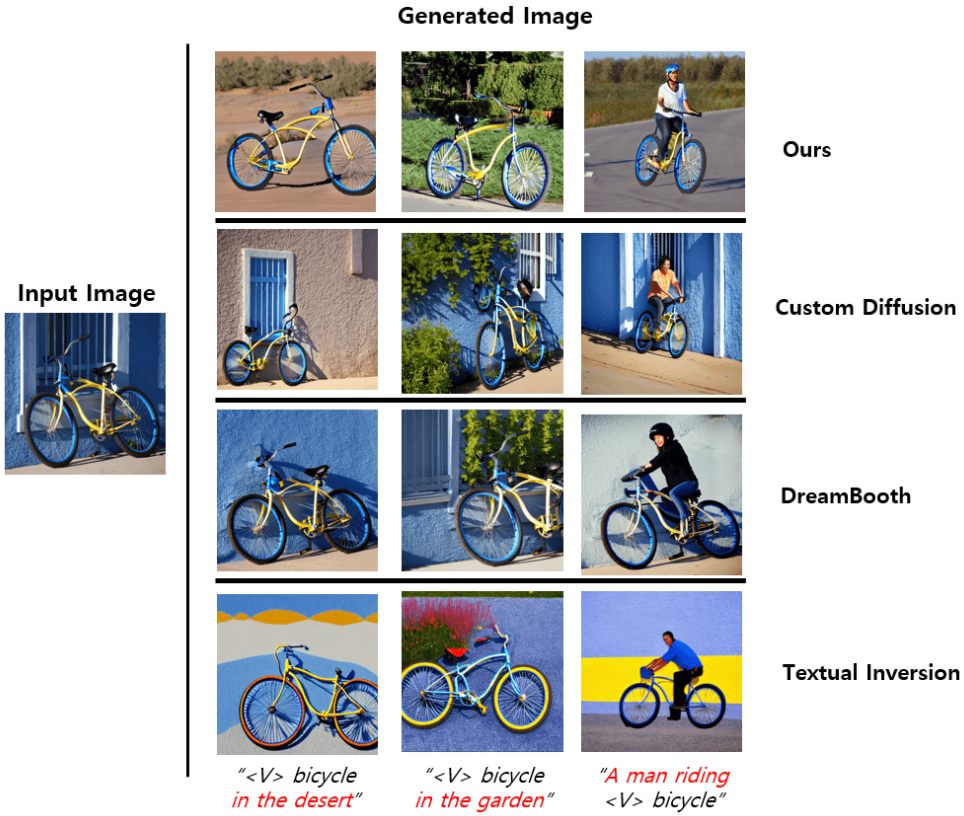

Figure 5. Comparison with previous methods that did not employ SOD mask preprocessing. We compared our method with previous approaches [14–16] that did not remove the backgrounds from the input images wherein the bicycle is intended to be preserved as the concept. The earlier methods are affected by extraneous information beyond the concept, such as the colors and composition of walls and floors, regardless of the text prompts entered. In contrast, our preprocessing removes such information, resulting in the generation of images that better align with the given conditions.

InstantBooth [18] is similar to our method in that it uses mask-based image preprocessing. However, there is a difference in the process of detecting the concept object and generating masks. While InstantBooth employs entity segmentation models, our approach utilizes SOD models. SOD models may have lower detection capabilities for multiple objects compared to entity segmentation models, but they are more suited for identifying the most prominent object within an image and offer computational efficiency with faster processing speeds. These features are particularly advantageous for on-site applications

requiring immediate image editing. For example, when taking product photographs in a store and needing to edit them on the spot to meet customer requirements, the quick preprocessing provided by SOD models presents a significant benefit.

3.3. Concept Embedding to Text Prompt

We inject the features of the concept image into the textual embedding to obtain a new text embedding. These new text embeddings are created by combining the image embedding of the concept image with the additional local features obtained from the concept image, replacing the embedding corresponding to the identifier. Our approach has demonstrated an improved capability to preserve the details of the concept over the previous methods [14–16] that employ random initialization of word embeddings corresponding to the identifier.

Initially, for a given target concept image for personalization, the text prompt must be modified accordingly. By drawing inspiration from DreamBooth [14], we inserted a unique identifier before the class noun to avoid the overhead caused by detailed descriptions of the concept image set. For instance, suppose the modified prompt takes the following form: 'A photo of $[V]$ chair', where '$[V]$' serves as the unique identifier and 'chair' represents the class noun. We denote this modified text prompt as $\hat{p}$ and utilize the CLIP text encoder to generate a text embedding, denoted as $CLIP_{Text}(\hat{p}) = \mathbf{f_{\hat{p}}}$. Subsequently, as depicted in Figure 6, by leveraging the pretrained CLIP image encoder, we obtain a feature vector, $\mathbf{f_k} = CLIP_{Image}(I_c)$, for the concept image I_c. Additionally, to utilize the local features of the concept image, which capture specific regions, patterns, and structures within the image, we extracted the local features $\mathbf{f_l}$ of the concept using a CNN. Specifically, local features containing features on shapes and forms were extracted from the pretrained ResNet-50 [20]. In the initial blocks of ResNet-50, low-level features, such as edges and corners, are output, whereas the intermediate blocks extract mid-level features representing patterns, structures, and forms. The latter stages handle more complex and abstract high-level features. Therefore, we used the local features, $\mathbf{f_l}$, of the concept extracted from the intermediate blocks of ResNet-50 to obtain additional information for preserving details, such as the structure, form, and patterns of the concept image. The image embedding, $\mathbf{f_k}$, and local feature, $\mathbf{f_l}$, are concatenated to form a new image embedding, $\mathbf{f}_c = \text{Concat}(\mathbf{f_k}, \mathbf{f_l})$. In order to prevent the influence of $\mathbf{f_l}$ from becoming dominant, normalization and rescaling were performed based on $\mathbf{f_c}$. Then, we found the embedding corresponding to the identifier, $[V]$, within $\mathbf{f_{\hat{p}}}$ and replaced it with $\mathbf{f_c}$. In this process, a trainable, fully connected (FC) layer is introduced to match the dimensions (768 dimension) of the text embeddings with the new image embeddings $\mathcal{T}$. The resulting finalized text embedding is used as a conditional input for the diffusion model, and fine-tuning is conducted through attention operations with the concept image at the cross-attention layer of the pretrained Diffusion U-net. Figure 7 visually demonstrates that the pattern of the concept image is more accurately maintained by applying the new image embedding technique that incorporates the local feature that we have proposed.

Our approach shares similarities with ELITE [17] and InstantBooth [18] in that it uses an image encoder to map visual features into the textual space. However, the key distinction lies in obtaining the visual feature to replace the text embedding. In contrast to the approaches used by ELITE and InstantBooth, our method utilizes the intermediate blocks of a pretrained ResNet-50 as a local feature extractor, combining the extracted local features with image embeddings to create new image embeddings that incorporate these local features. These are then mapped to the word embeddings corresponding to the identifiers that serve as conditions for the diffusion process.

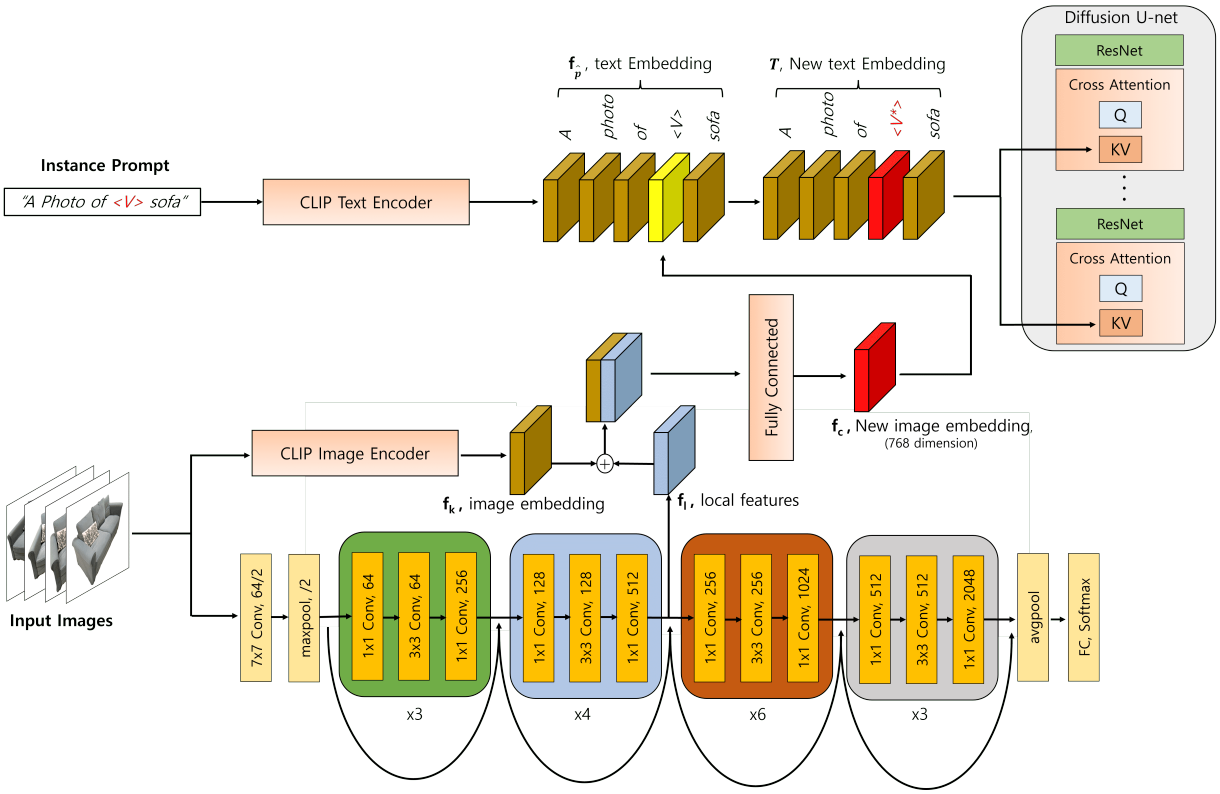

Figure 6. Our network structure to obtain new image embeddings, including local features. The concept image is used as input to obtain image embeddings through a pretrained CLIP image encoder. In addition, local features are extracted from the intermediate blocks of a pretrained ResNet-50 and are combined with the image embeddings. A trainable Fully Connected (FC) layer is introduced to align the dimensions with the text embeddings, from which new image embeddings are derived. These new image embeddings replace the identifier text embeddings and are utilized as the condition for fine-tuning in the diffusion process.

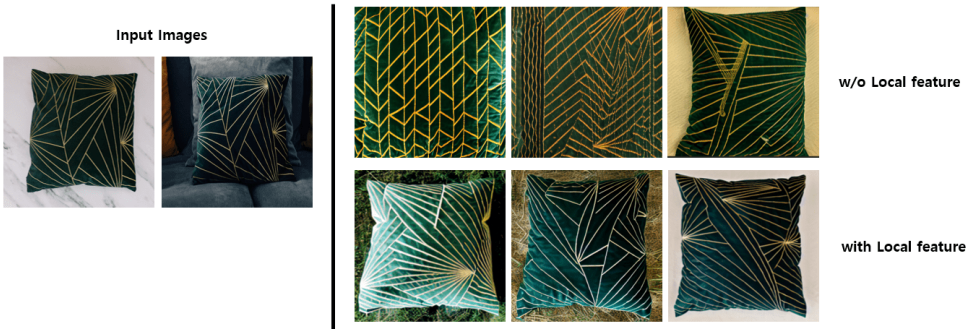

Figure 7. Visual comparison of the results for the new image embedding with the local feature. By additionally providing the local feature, it can be confirmed that the details of the pattern in the input image are better preserved.

3.4. Image Similarity Measurement Using Siamese Networks

In postprocessing, we selected the final image by assessing the similarity between two generated images. We employed a Siamese neural network [21] designed to quantify image similarity. This model is trained to distinguish between pairs of images and calculate a similarity score. The Siamese network has two identical subnetworks, each taking individual input images and producing corresponding feature vectors. The final similarity score is computed based on the Euclidean distance between these feature vectors. Siamese network

training uses a contrastive loss function tailored to determine whether two input samples are similar or dissimilar. This loss is defined as

$$L_{con} = (1 - Y) \times \frac{1}{2} D^2 + Y \times \frac{1}{2} \{max(0, M - D)\}^2, \quad (4)$$

where Y represents a binary class variable that signifies whether the image pair belongs to the same class (1) or different classes (0). Variable D is the Euclidean distance between the feature vectors of image pairs produced by the Siamese network, and M is the hyperparameter that serves as a margin to determine the desired separation between embeddings for different pairs. We trained the Siamese network by pairing concept images with images from different objects within the same class. Subsequently, we used images generated by diffusion alongside concept images to rank them based on similarity scores represented by Euclidean distance. We removed the bottom 40% of images to filter out those images with insufficient similarity, resulting in the final sample. Figure 8 illustrates the results of the similarity scores computed using the Siamese network.

Figure 8. Image samples based on similarity scores. When using the Siamese network, we measure the similarity score between the concept and generated images, selecting only those with high similarity. The similarity score is computed by calculating the Euclidean distance between the embeddings of the two images, where a low score indicates high similarity. Samples with scores in the bottom 40% were classified as negative images, and the remaining samples were designated as positive images.

3.5. Model Training

3.5.1. Training Loss

We utilized two types of loss functions during training.

First, L_{LDM} learns the latent representation of the input image from a noise vector, enabling the effective reconstruction of complex textures and details in the image. It also incorporates text prompts conditionally to properly adjust the relationship between text and image. Here, instead of using parameter y for the conditional input, we used the text embedding $\mathcal{T}$ newly obtained from the image embedding and local feature. The newly defined L_{LDM} is as follows:

$$L_{LDM} := \mathbb{E}_{z \sim \varepsilon(x), y, \epsilon \sim \mathcal{N}(0,1), t}[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(\mathcal{T}))\|_2^2], \quad (5)$$

Second, we introduced a cosine similarity loss to train the FC layer appended to the image encoder and ResNet, which was used as a local feature extractor. While L_{LDM} focuses on the accuracy of image reconstruction, the cosine similarity loss L_{cos} is centered on enhancing the similarity between the embeddings of the generated images and the embeddings of the concept images. The cosine similarity loss, which is generally used as a traditional loss function, is being applied in various studies. Barz et al. [48] demonstrated the usefulness of cosine loss in maximizing performance, especially in cases of small datasets with a limited amount of training data. Our method, which utilizes only 4 to 8 input images during the fine-tuning process, introduces cosine loss to maximize the similarity between the concept images and the generated images. Moreover, SimCLR [49] measured the cosine similarity between the image embeddings of two images to assess the similarity of the images. SimCLR calculated the cosine similarity by including both similar and dissimilar images in the learning process for training regarding the judgment of similarity between the selected images and then defined the loss function by applying it to a softmax. However, our study focuses solely on ensuring that the generated images are similar to the concept images, thereby only measuring the cosine similarity between the embeddings of the concept image and the generated image, and the cosine similarity loss is as follows:

$$L_{cos} = 1 - \frac{f_c \cdot f_g}{\|f_c\| \|f_g\|}, \tag{6}$$

where f_c represents the concept image embedding concatenated with local feature, and f_g is the image embedding of the generated sample. When the embeddings are highly similar, the loss approaches 0, indicating a closer match between the generated sample and concept image. Conversely, the loss increases as the embeddings diverge. We defined the total loss function by combining these two loss functions, and we applied it during the training process. Through experimentation, we found that if the proportion of the cosine loss is excessively large, it yields good results in stylization but fails to properly reflect the text prompts in editing. Conversely, if the proportion is too small, the opposite occurs. In order to solve this issue, we introduced a learnable parameter, α, to apply an appropriate ratio of cosine loss, thereby determining the final cosine loss. The initial value of α is set to 0.5, and the overall loss is as follows:

$$L = L_{LDM} + \alpha L_{cos} \tag{7}$$

We updated the FC layer that was additionally connected for image embeddings, the ResNet for local feature extraction, and only the Key and Value weights of the cross-attention layer of the diffusion network, as it has been shown in Custom Diffusion [15] that updating only the Key and Value weights of the cross-attention layer is sufficient to improve the model's understanding of text-image pairs. The rest are frozen.

The training process is shown in Figure 9, and the effects of L_{cos} are described in Section 4.3.2.

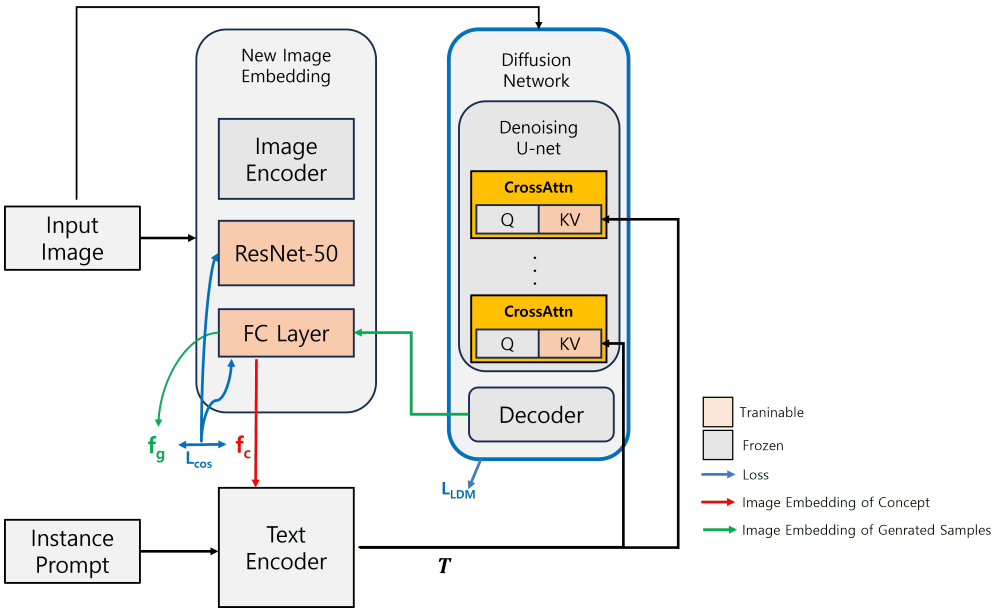

Figure 9. The workflow for calculating our model's loss function.

3.5.2. Results Per Training Epochs

In order to verify the performance of our method, we examined the visual outcomes generated at each stage of the generation process and measured the image alignment, text alignment, and DINO scores, epoch by epoch. As seen in Figure 10, the cup images, which have relatively simple patterns and structures in the input image, are well-represented by both the baseline model [15] and our model. However, the baseline tends to over-imitate the input image, causing overfitting and an inability to properly follow the prompt. Conversely, our method performed well, as it was prompted without being affected by the background, demonstrating the effectiveness of the preprocessing method. Similarly, for the clock input images at the bottom, our method was less affected by background noise and better captured the color and structure of the clock compared to the baseline model. This improvement appears to originate from the additional local features learned that provided color and pattern details for the concept image. Moreover, implementing cosine similarity loss during training seems to have effectively preserved the concept by enhancing the similarity between the embeddings of the generated samples and the concept image. Moreover, Figure 11 illustrates how image alignment, text alignment, and DINO alignment change over the epochs. As the training progresses, the gradual increase in image alignment and DINO scores indicates effective learning in preserving the concept image. Furthermore, the improvement in text alignment demonstrates that the generated images are being trained to faithfully follow the text prompts.

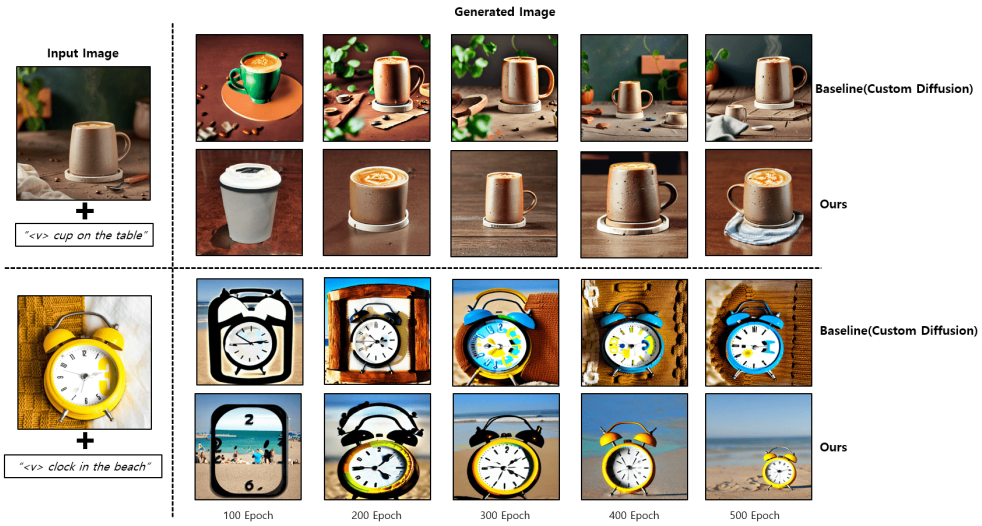

Figure 10. Epoch-by-epoch visual results compared to the baseline.

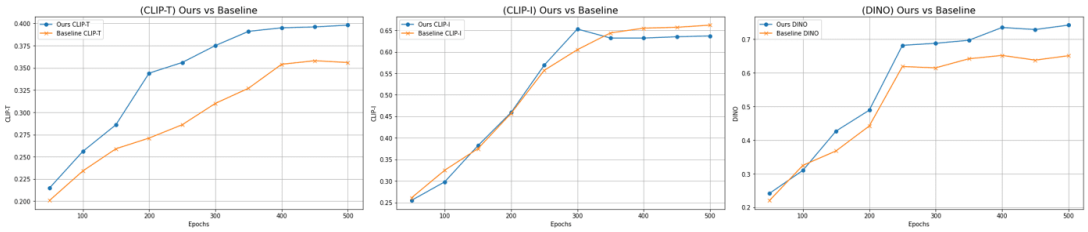

Figure 11. Quantitative score evolution with training epochs.

4. Experiment

We present the datasets and evaluation metrics used in our experiments, and subsequently compare and analyze the results obtained using our method with those of existing approaches.

4.1. Dataset and Evaluation

4.1.1. Datasets

We conducted experiments on 10 target datasets, encompassing various categories, including furniture items, such as chairs, tables, beds, and sofas, as well as animals, such as dogs and cats.

The images used for the experiments were sourced from the BIG [50] dataset, which includes high-resolution images, ranging from 2048 × 1600 to 5000 × 3600, from Unsplash [51], which is known for providing copyright-free high-resolution images. In addition, we extracted class-specific images from the large-scale text-image dataset LAION-400M [52], utilizing these as the regularization dataset and also for training the Siamese network as either positive or negative datasets.

Furthermore, to enhance the reliability of our experiments, we utilized images that we had captured ourselves. This approach played a significant role in assessing the network's performance compared to existing datasets and in verifying the applicability of our research findings.

Figure 12 showcases one or more sample images for each subject. Figure 13 shows the results of the experiments with our own dataset. In order to evaluate general performance, we compared the generation results of our method to those of previous methods using self-captured data at resolutions below 1000 × 1000. For the cat toy (left), both Custom Diffusion and DreamBooth were influenced by background information and failed to generate accurate images corresponding to the prompt. Moreover, Textual Inversion produced completely different images, but our method represented the cat's face and the context

of the prompt relatively well. Similarly, for the flowerpot (right), the previous methods were affected by the background area, or the number and shape of black labels differed from the input image. Our method, however, accurately depicted the location and number. Nevertheless, we observed that there are still limitations in depicting very fine details, such as the content of the text.

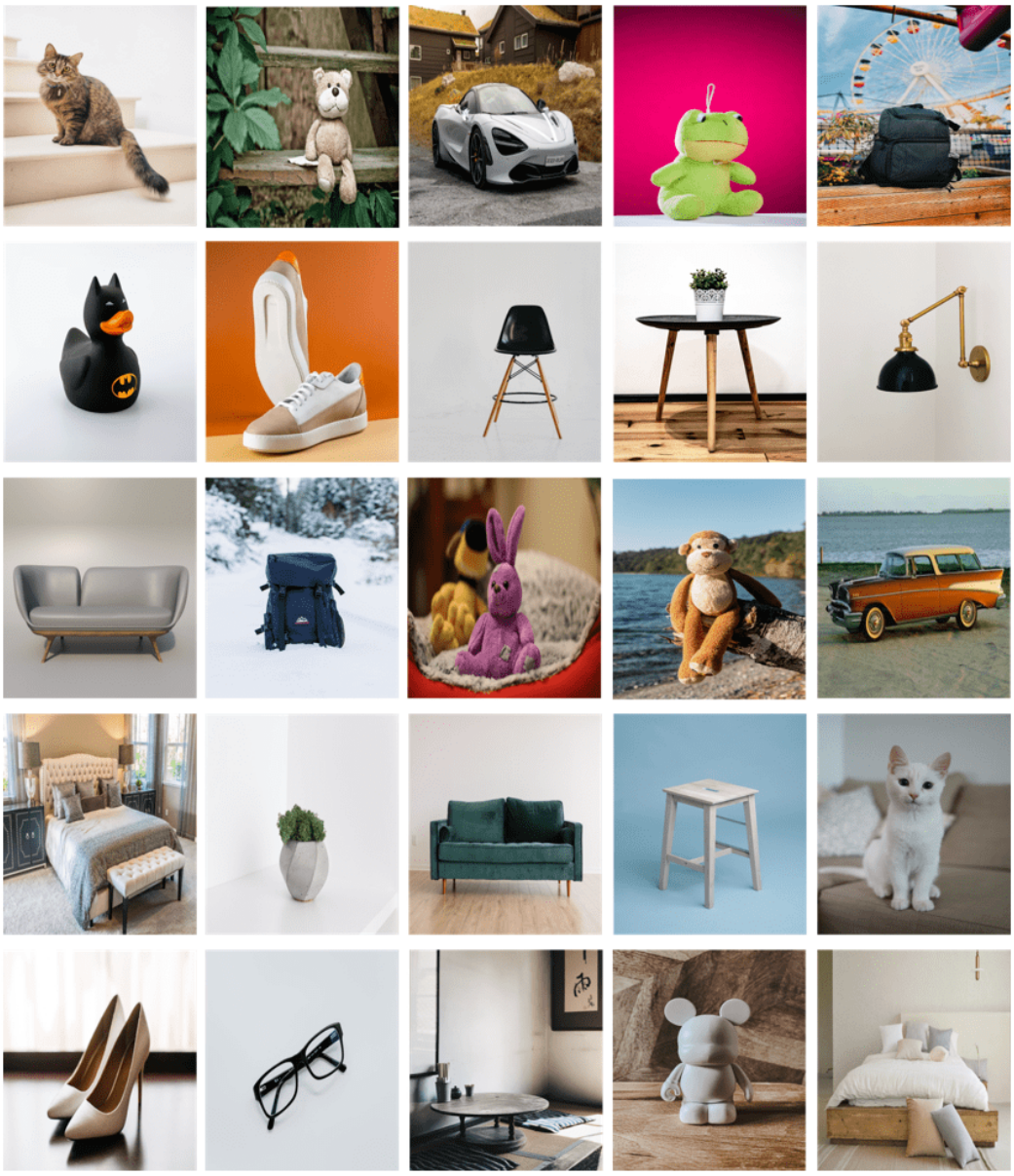

Figure 12. The dataset, with over 10 categories.

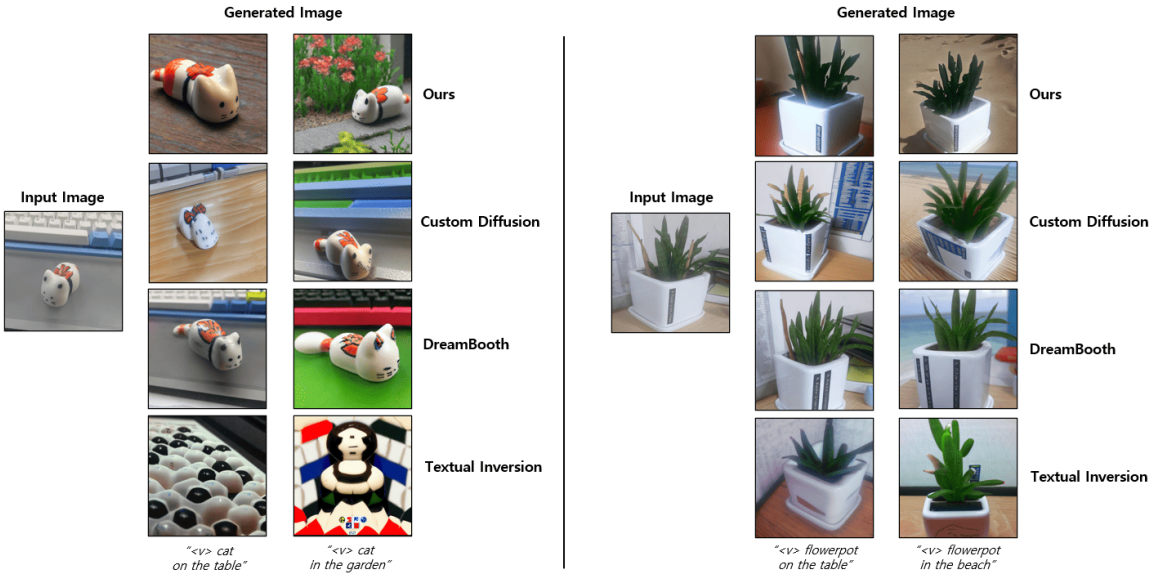

Figure 13. Comparison between the generated results (using our own data) and those of the previous methods.

4.1.2. Evaluation Metrics

We employed several metrics to evaluate the fidelity of the generated images, the distributional similarity between the concept and generated images, and the alignment between the given prompt and the image. First, we evaluated the similarity between the generated and actual images using CLIP [22] image alignment and DINO [53] evaluation metrics. CLIP image alignment measures pairwise cosine similarity between the embeddings of the generated and real images, reflecting their semantic content alignment. In contrast, DINO assesses the cosine similarity between the embeddings of an image, focusing on the fidelity and distinctiveness of the features and structures within the image. In other words, CLIP image alignment quantifies the 'similarity' of the content of two images, while DINO helps distinguish the detailed differences between images or objects within the same class. Additionally, we calculate the kernel inception distance (KID) [54] between the generated and concept images to measure distributional similarity. Furthermore, CLIP text alignment is computed to assess the alignment between the given prompt and image by measuring the average cosine similarity between the prompt and image embeddings. The results of these measurements are given in Table 1. Our method shows an increase in the CLIP-T and DINO metrics compared to the previous methods. This suggests that the images generated by our approach more closely follow the given prompts. We surmise that this results from the preprocessing technique we introduced, which reduces background interference. The increase in DINO also indicates an improved ability to preserve objects and patterns within the image. However, we have observed a decrease in the CLIP-I and KID metrics compared to Custom Diffusion [15]. This can be attributed to the fact that the resulting images generated by the previous methods retain more of the input image's background, which is also included in the concept images that are the subject of measurement.

Table 1. Quantitative evaluation comparison. CLIP image alignment and CLIP text alignment are denoted as CLIP-I and CLIP-T, respectively. Compared to existing models, we observed improvements in the metrics according to CLIP-T and DINO [53]. Our proposed approach more accurately represents images related to the prompt's description and discerns finer details within the images.

Method	CLIP-T ↑	CLIP-I↑	DINO↑	KID↓
Textual Inversion	0.259	0.586	0.428	32.2
DreamBooth	0.421	0.785	0.654	16.7
Custom Diffusion	0.440	**0.882**	0.666	**8.21**
Ours	**0.460**	0.852	**0.743**	8.83

Higher values for CLIP-T, CLIP-I, and DINO indicate better performance, while lower values for KID signify superior performance. Values highlighted in bold indicate the best performance.

4.2. Implementation Details

We employed Stable Diffusion [1] v1-4 as a pretrained [55], large-scale text-to-image model for the experiment. For image embeddings, we employed the CLIP image encoder with an additional FC layer. To extract local features, we used a CNN up to the intermediate layers of a pretrained ResNet, which was our local feature extractor. During training, all parameters were frozen except for the diffusion cross-attention layer, the FC layer of the image encoder, and the CNN local feature extractor. In order to optimize the image encoder and CNN, we incorporated a cosine similarity loss, parameterized by α, which was initialized at 0.5 and constrained not to exceed 1. During data preprocessing, we superimposed the salient object extracted using the SOD mask onto various monochromatic backgrounds. This strategy accentuated the prominence of object information amidst monotonous backgrounds. We set the batch size to 4 for our training configurations and adapted the number of training steps based on categories. Specifically, objects, such as pieces of furniture (e.g., chairs and tables), exhibiting complex patterns or an inconsistent number of legs were subjected to more than 500 training steps. Conversely, concepts with more straightforward forms underwent 250 steps. We trained the networks using a learning rate of 1×10^{-5} on an Intel Core i7-10700 processor clocked at 2.9 GHz, with two NVIDIA RTX 3090 GPUs, each equipped with 24 GB GPU memory.

4.3. Ablation Study

In this section, we conduct an ablation experiment to evaluate the effect of SOD mask preprocessing and cosine similarity loss.

4.3.1. Preprocessing Using SOD Mask

We conducted ablation experiments to compare the effects of preprocessing with and without the use of SOD masks, and we analyzed both the qualitative and quantitative results.

Figure 14 compares the visual results obtained using the masked and unmasked images. When SOD mask preprocessing is not applied, it can be observed that the generated images do not preserve the concept well. This could be inferred as the result of the features from objects other than the concept object blending into the target object.

Additionally, a qualitative evaluation was conducted for whether or not the SOD mask was used. In Table 2, CLIP image alignment and DINO exhibited the most significant improvements when employing the SOD mask. By filtering unnecessary information from the concept image using the SOD mask, the generated samples exhibit a higher fidelity to the concept image.

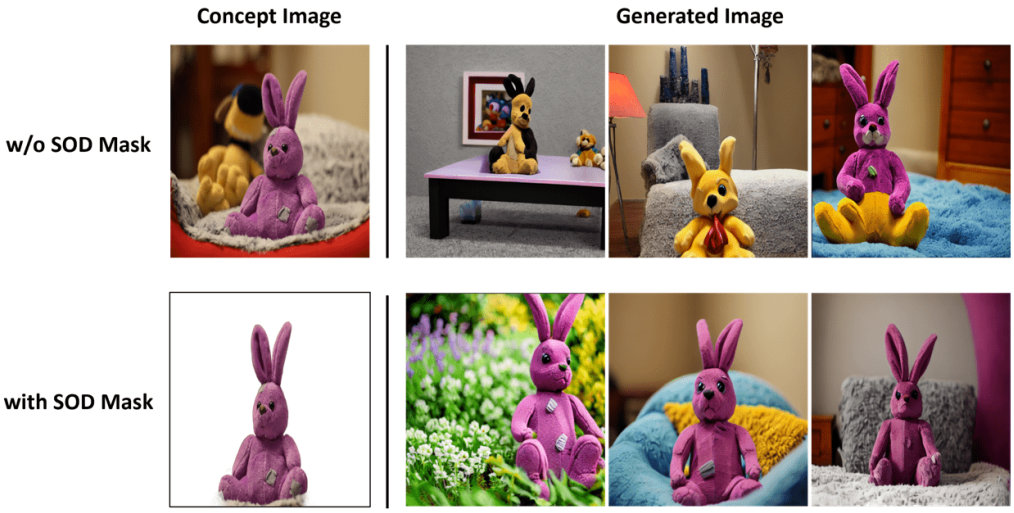

Figure 14. Generated image after removing unnecessary regions using SOD model [19]. When using the concept image without preprocessing, the desired object could not be properly preserved due to unwanted features being present in the background, such as the yellow entity (**first row**). In contrast, using the SOD model to detect only the salient objects, the concept remains unaffected by the background (**second row**).

Table 2. Quantitative comparison for SOD mask ablation.

Method	CLIP-T ↑	CLIP-I↑	DINO↑	KID↓
w/o SOD mask	0.415	0.657	0.572	18.2
with SOD mask	**0.424**	**0.763**	**0.691**	**14.5**

Higher values for CLIP-T, CLIP-I, and DINO indicate better performance, while lower values for KID signify superior performance. Values highlighted in bold indicate the best performance.

4.3.2. Cosine Similarity Loss Ablation

Next, we compare the model performance based on the cosine similarity loss introduced to train the FC layer appended to the image encoder for text embedding conversion and the CNN local feature extractor. Figure 15 displays samples based on the cosine similarity loss, and Table 3 presents a quantitative comparison.

Table 3. Quantitative comparison using cosine similarity loss with the learnable parameter α. We conducted a qualitative comparison based on the influence ratio of L_{cos}. The optimal parameter value of 0.6 obtained through learning provided the best results in CLIP text and image alignment and KID [54]. When α is high, the DINO evaluation showed favorable results but had the lowest text alignment. As α increases, DINO captures details better, but overfitting results in a less accurate alignment with the textual description.

Method	CLIP-T ↑	CLIP-I↑	DINO↑	KID↓
w/oL_{cos}	0.672	0.671	0.418	6.85
$\alpha = 0.6$	**0.677**	**0.698**	0.412	**6.34**
$\alpha = 6$	0.655	0.693	**0.455**	7.65

Higher values for CLIP-T, CLIP-I, and DINO indicate better performance, while lower values for KID signify superior performance. Values highlighted in bold indicate the best performance.

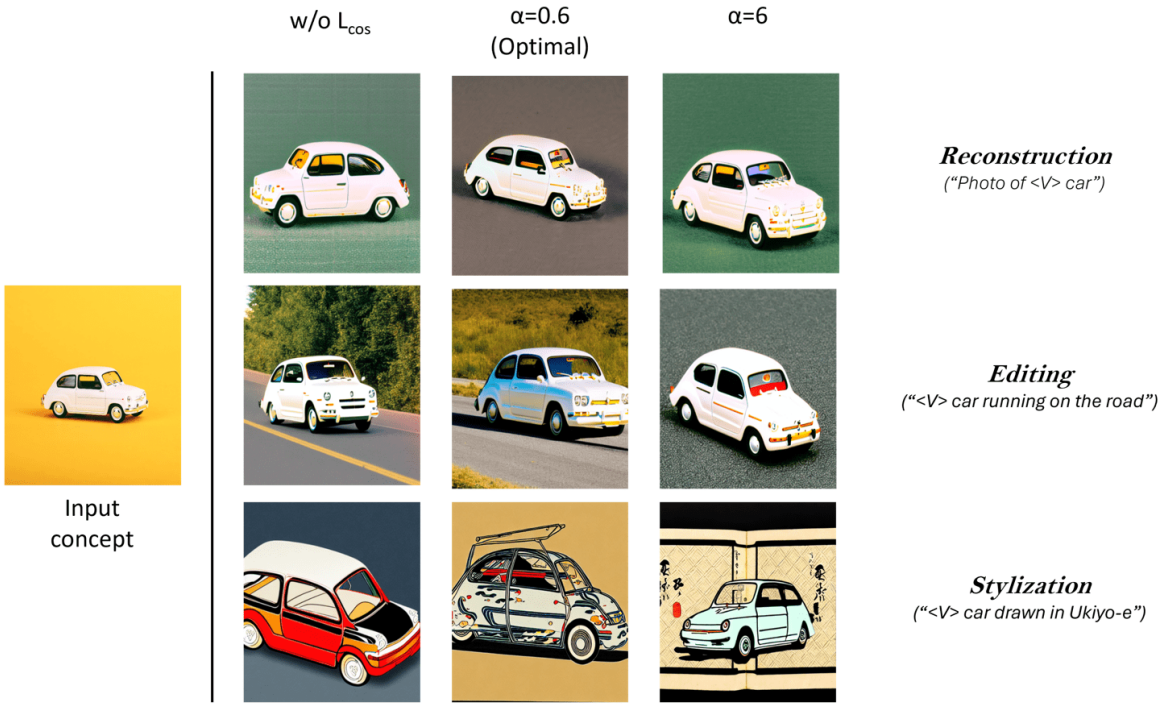

Figure 15. Results based on cosine similarity loss L_{cos} and the trainable parameter α. We compared the result samples of Reconstruction (**first row**), Editing (**second row**), and Stylization. For the reconstruction samples, good results were obtained regardless of the L_{cos} value. In addition, appropriate results for the text prompt were produced in Editing and Stylization for small and large values of L_{cos}, respectively. Moreover, a balanced result is observed at $\alpha = 0.6$.

4.4. Qualitative Results

In this section, we compare the visual results of our approach with those of existing models.

Visual Comparison

In Figure 16, we use the same prompts to compare the proposed method with existing methods for image reconstruction, image editing, and style transfer across five categories (such as sofa, toys, and vase). Notably, Textual Inversion [16] distorts information, such as ratios and colors, and may not entirely adhere to the prompts. DreamBooth generates high-quality images but has substantial training time and storage requirements. In contrast, Custom Diffusion enhances speed but cannot incorporate fine patterns and structural information. Our method shares a similar speed to Custom Diffusion, yet it mitigates the loss of color and structural information while preserving detailed patterns.

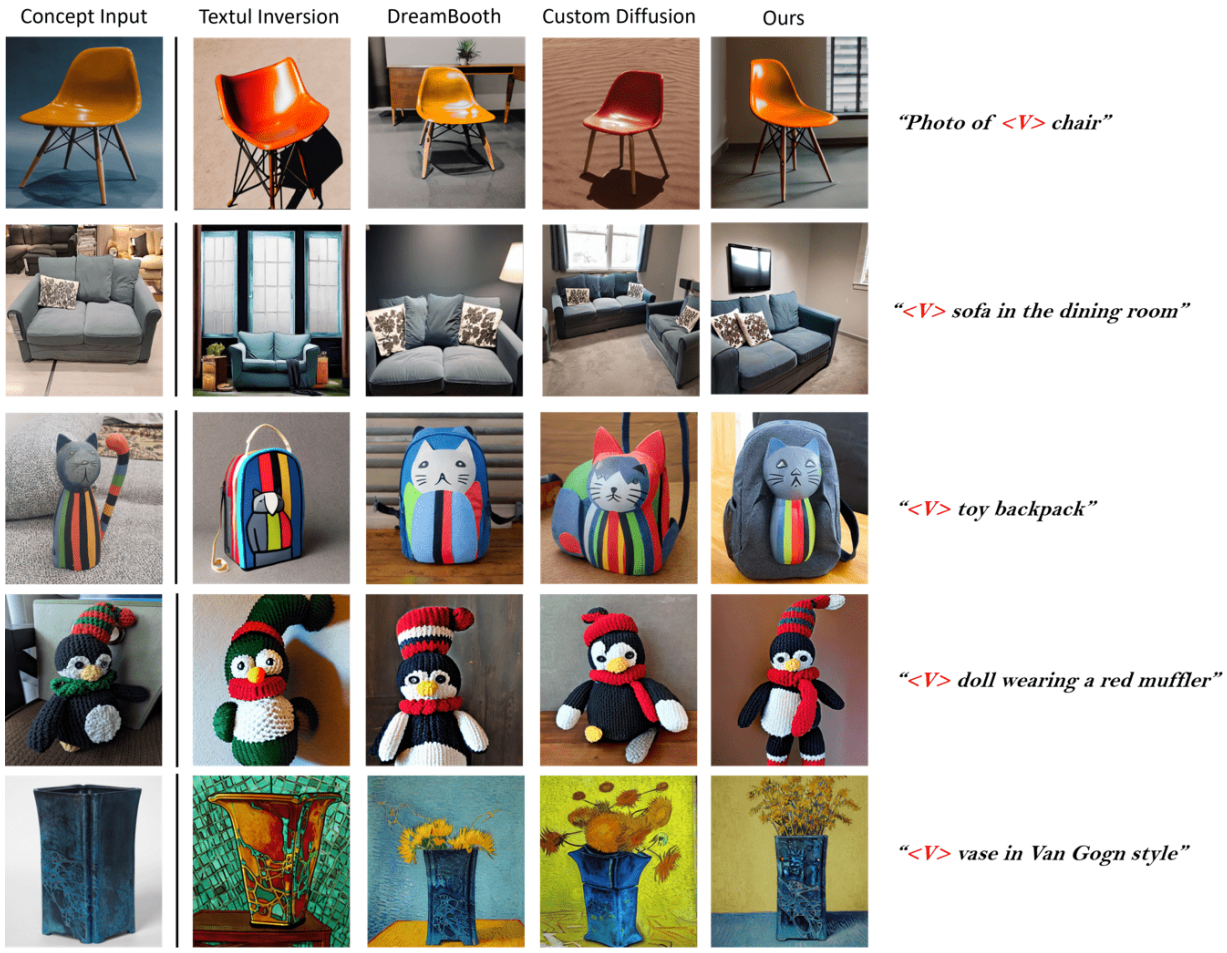

Figure 16. Visual comparison of existing methods. The generated outcomes that had given concept images as input (**first column**). We visually compared the results of image reconstruction (**first row**), image editing (**rows 2**, **3**, **and 4**), and style transfer (**fifth row**) generated under the same prompts between the existing methods [14–16] (**columns 2**, **3**, **and 4**) and our approach (**fifth column**). The proposed method better preserves the detailed patterns and structures of the cat's face (**third row**), the color and pattern of the penguin toy (**fourth row**), and the intricate pattern and structure of the vase (**fifth row**).

4.5. Failure Cases

Our method occasionally over-emphasizes patterns or produces incorrect structures when faced with excessively intricate patterns, complex structures, or situations where parts of the concept object are occluded. Figure 17 illustrates the failed generation results, which make it difficult to discern its complete form.

Figure 17. Failure cases. Limitations in the image quality can be observed when dealing with irregular structures (**first row**) or fine patterns (**second row**). Additionally, the results are inaccurate for objects with occlusion (**third row**).

5. Discussion and Conclusions

We introduced a method to enhance the performance of personalized image generation models trained on large-scale text-to-image models, aiming to improve detail preservation in personalized text-to-image tasks. Our experiments used high-resolution image datasets collected from sources like BIG and Unsplash, as well as images captured by ourselves. By providing clear information about the region of the image to be preserved and using the SOD mask, we reduce unnecessary background information that contaminates the generated output. Unlike similar studies that preprocess using entity segmentation, preprocessing with SOD aims to detect only the most prominent objects, allowing for efficient preprocessing at a faster rate. This enabled us to observe visual improvements compared to previous methods that were unable to follow the input text prompts properly due to interference from unnecessary areas like the background, and we also noted enhancements in text alignment. Additionally, mapping the concept image to the text embedding space allowed for the utilization of a wealth of visual information. We mapped the concept image to the text space using pretrained image and text encoders. In contrast to prior research, our method employs a CNN local feature extractor to supply local features in conjunction with text embeddings. The incorporated local features offer a wealth of information on the patterns, colors, and structures of the concept image, generating new image embeddings mapped to the text space. This helped ensure the generated images better preserve the concept's details, as confirmed by the visual results and improvements in qualitative metrics like DINO. Furthermore, the introduction of cosine similarity loss guided the generation of images that were more similar to the input concept image. Although there was a slight decrease in CLIP Image alignment and KID compared to the baseline model, we attribute this to the tendency of previous methods to overfit to the input image, particularly when the input is ambiguous due to the background. Finally, during postprocessing, we employed a Siamese network to selectively choose high-similarity images. Our strategies demonstrated the generation of images with high fidelity that closely follow the prompts. However, we

also recognized the generative limitations in creating very fine patterns, the text included in an image, and unusual structures and proportions. Although our experiments utilized fomulas that are established and traditional, future work may benefit from incorporating more advanced, precise techniques to further enhance the outcomes. Moving forward, we plan to explore multimodal methods in future research to improve these issues by employing additional conditions beyond text prompts.

Author Contributions: Conceptualization, J.Y. and S.K.; Methodology, M.K.; software, M.K.; Validation, S.K. and M.K.; Investigation, M.K. and S.K.; Writing—original draft preparation, M.K.; Writing—review and editing, S.K.; Supervision, J.Y. and S.K.; Project administration, J.Y. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the ICT R&D program of MSIT/IITP[RS-2022-0015604, Development of object/drawing technology and web application for augmented reality interior service]. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICAN (ICT Challenge and Advanced Network of HRD) program (IITP-RS-2022-00156215) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation). The present research was conducted under the Research Grant of Kwangwoon University in 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022 ; Volume 10, pp. 10684–10695.
2. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125.
3. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 36479–36494.
4. Wang, P.; Yang, A.; Men, R.; Lin, J.; Bai, S.; Li, Z.; Ma, J.; Zhou, C.; Zhou, J.; Yang, H. OFA: Unifying Architectures, Tasks, and Modalities through a Simple Sequence-to-Sequence Learning Framework. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 23318–23340.
5. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5907–5915.
6. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
7. Zhang, Y.; Huang, N.; Tang, F.; Huang, H.; Ma, C.; Dong, W.; Xu, C. Inversion-Based Style Transfer with Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 10146–10156.
8. Balaji, Y.; Nah, S.; Huang, X.; Vahdat, A.; Song, J.; Kreis, K.; Liu, M.Y. eDiffi: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers. *arXiv* **2022**, arXiv:2211.01324.
9. Tumanyan, N.; Geyer, M.; Bagon, S.; Dekel, T. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 1921–1930.
10. Sohn, K.; Ruiz, N.; Lee, K.; Chin, D.C.; Blok, I.; Chang, H.; Krishnan, D. StyleDrop: Text-to-Image Generation in Any Style. *arXiv* **2023**, arXiv:2306.00983.
11. Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; Cohen-Or, D. Prompt-to-Prompt Image Editing with Cross Attention Control. *arXiv* **2022**, arXiv:2208.01626.
12. Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Irani, M. Imagic: Text-based Real Image Editing with Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 6007–6017.
13. Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Chen, M. Glide: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. *arXiv* **2021**, arXiv:2112.10741 .

14. Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; Aberman, K. Dreambooth: Fine Tuning Text-to-Image Diffusion Models for Subject-driven Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 22500–22510.
15. Kumari, N.; Zhang, B.; Zhang, R.; Shechtman, E.; Zhu, J.Y. Multi-concept Customization of Text-to-Image Diffusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 1931–1941.
16. Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A.H.; Chechik, G.; Cohen-Or, D. An Image is Worth One Word: Personalizing Text-to-Image Generation Using Textual Inversion. *arXiv* **2022**, arXiv:2208.01618.
17. Wei, Y.; Zhang, Y.; Ji, Z.; Bai, J.; Zhang, L.; Zuo, W. Elite: Encoding Visual Concepts into Textual Embeddings for Customized Text-to-Image Generation. *arXiv* **2023**, arXiv:2302.13848.
18. Shi, J.; Xiong, W.; Lin, Z.; Jung, H.J. Instantbooth: Personalized Text-to-Image Generation without Test-Time Finetuning. *arXiv* **2023**, arXiv:2304.03411 .
19. Kim, T.; Kim, K.; Lee, J.; Cha, D.; Lee, J.; Kim, D. Revisiting Image Pyramid Structure for High Resolution Salient Object Detection. In Proceedings of the Asian Conference on Computer Vision, Macau, China, 4–8 December 2022; pp. 108–124.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
21. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese Neural Networks for One-Shot Image Recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015; Volume 2, No. 1.
22. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sutskever, I. Learning Transferable Visual Models from Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 8748–8763.
23. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative Adversarial Text to Image Synthesis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1060–1069.
24. Zhu, M.; Pan, P.; Chen, W.; Yang, Y. DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5802–5810.
25. Liao, W.; Hu, K.; Yang, M.Y.; Rosenhahn, B. Text to Image Generation with Semantic-Spatial Aware GAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18187–18196.
26. Gal, R.; Patashnik, O.; Maron, H.; Bermano, A.H.; Chechik, G.; Cohen-Or, D. StyleGAN-NADA: CLIP-guided Domain Adaptation of Image Generators. *Acm Trans. Graph. (TOG)* **2022**, *41*, 1–13. [CrossRef]
27. Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; Lischinski, D. StyleCLIP: Text-driven Manipulation of StyleGAN Imagery. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2085–2094.
28. Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Tang, J. Cogview: Mastering Text-to-Image Generation via Transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 19822–19835.
29. Chang, H.; Zhang, H.; Barber, J.; Maschinot, A.J.; Lezama, J.; Jiang, L.; Krishnan, D. Muse: Text-to-Image Generation via Masked Generative Transformers. *arXiv* **2023**, arXiv:2301.00704.
30. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-Shot Text-to-Image Generation. In Proceedings of the 38th International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; Voume 139, pp. 8821–8831.
31. Zhang, Z.; Han, L.; Ghosh, A.; Metaxas, D.N.; Ren, J. Sine: Single Image Editing with Text-to-Image Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 6027–6037.
32. Li, Y.; Liu, H.; Wu, Q.; Mu, F.; Yang, J.; Gao, J.; Li, C.; Lee, Y.J. GLIGEN: Open-Set Grounded Text-to-Image Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 22511–22521.
33. Zhang, L.; Rao, A.; Agrawala, M. Adding Conditional Control to Text-to-Image Diffusion Models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Vancouver, BC, Canada, 18–22 June 2023; pp. 3836–3847.
34. Makhmudkhujaev, F.; Hong, S.; Park, I.K. Re-Aging GAN: Toward Personalized Face Age Transformation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 3908–3917.
35. Kim, H.; Choi, Y.; Kim, J.; Yoo, S.; Uh, Y. Exploiting Spatial Dimensions of Latent in GAN for Real-Time Image Editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 852–861.
36. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
37. Hou, Q.; Cheng, M.-M.; Hu, X.; Borji, A.; Tu, Z.; Torr, P.H.S. Deeply Supervised Salient Object Detection With Short Connections. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3203–3212.
38. Xie, S.; Tu, Z. Holistically-Nested Edge Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1395–1403.

39. Xie, C.; Xia, C.; Ma, M.; Zhao, Z.; Chen, X.; Li, J. Pyramid Grafting Network for One-Stage High Resolution Saliency Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11717–11726.
40. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
41. Pang, Y.; Zhao, X.; Zhang, L.; Lu, H. Multi-Scale Interactive Network for Salient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9413–9422.
42. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
43. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
44. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
45. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; IEEE: New York, NY, USA, 2011; pp. 2564–2571.
46. Hoffer, E.; Ailon, N. Deep Metric Learning Using Triplet Network. In *Similarity-Based Pattern Recognition, Proceedings of the Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, 12–14 October 2015;* Proceedings 3; Springer International Publishing: Berlin/Heidelberg, Germany, 2015.
47. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
48. Barz, B.; Denzler, J. Deep learning on small datasets without pre-training using cosine loss. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1371–1380.
49. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning Machine Learning, PMLR, Virtual, 13–18 June 2020; pp. 1597–1607.
50. Cheng, H.K.; Chung, J.; Tai, Y.W.; Tang, C.K. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8890–8899.
51. Unsplash. Available online: https://unsplash.com/ (accessed on 8 November 2023).
52. Schuhmann, C.; Vencu, R.; Beaumont, R.; Kaczmarczyk, R.; Mullis, C.; Katta, A.; Coombes, T.; Jitsev, J.; Komatsuzaki, A. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv* **2021**, arXiv:2111.02114.
53. Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9650–9660.
54. Bińkowski, M.; Sutherland, D.J.; Arbel, M.; Gretton, A. Demystifying mmd gans. *arXiv* **2018**, arXiv:1801.01401.
55. Stable-Diffusion v1-4. 2022. Available online: https://huggingface.co/CompVis/stable-diffusion-v-1-4-original (accessed on 11 November 2023).