electronics

MDPI

Article

Temporally Coherent Video Cartoonization for Animation Scenery Generation

Gustavo Rayo and Ruben Tous *

Department of Computer Architecture, Universitat Politecnica de Catalunya, 08034 Barcelona, Spain; gustavo.enrique.rayo@estudiantat.upc.edu
* Correspondence: ruben.tous@upc.edu; Tel.: +34-934054044

Abstract: The automatic transformation of short background videos from real scenarios into other forms with a visually pleasing style, like those used in cartoons, holds application in various domains. These include animated films, video games, advertisements, and many other areas that involve visual content creation. A method or tool that can perform this task would inspire, facilitate, and streamline the work of artists and people who produce this type of content. This work proposes a method that integrates multiple components to translate short background videos into other forms that contain a particular style. We apply a fine-tuned latent diffusion model with an image-to-image setting, conditioned with the image edges (computed with holistically nested edge detection) and CLIP-generated prompts to translate the keyframes from a source video, ensuring content preservation. To maintain temporal coherence, the keyframes are translated into grids and the style is interpolated with an example-based style propagation algorithm. We quantitatively assess the content preservation and temporal coherence using CLIP-based metrics over a new dataset of 20 videos translated into three distinct styles.

Keywords: diffusion probabilistic models; deep learning; animation; generative models; video stylization; video cartoonization; video translation; computer graphics

Citation: Rayo, G.; Tous, R. Temporally Coherent Video Cartoonization for Animation Scenery Generation. *Electronics* **2024**, *13*, 3462. https://doi.org/10.3390/electronics13173462

Academic Editors: Sergio Saponara and Abdussalam Elhanashi

Received: 6 August 2024
Revised: 28 August 2024
Accepted: 29 August 2024
Published: 31 August 2024

1. Introduction

There is a constant demand for the creation of new and visually captivating content in terms of images and videos. Cartoon-based videos have been an effective means of communication and entertainment, not only for children but also for people of all ages. However, the creative process has traditionally relied on the skill and creativity of artists, requiring a great deal of manual work. One common task where artists have to invest a significant amount of time is in the creation of assets such as animated backgrounds, which are usually inspired by real scenarios to provide a sense of realism and infuse cultural and geographical context. Their application extends beyond animated films; they could also be used in other areas, such as video games, advertisements, and even educational content.

Recent advances in Artificial Intelligence (AI) have revealed the possibility of automating this process to some extent. The latest progress in conditional generative models offers a promising approach to addressing this issue. Generative models such as GANs have produced impressive results in different tasks, such as style transfer [1], image-to-image translation [2,3], and even in video-to-video translation [4,5]. However, GANs are difficult to train and struggle to capture the full data distribution [6]. More recently, diffusion models [7] have gained popularity and can also be used in many tasks similar to GANs.

Motivated by the demand for tools in visual content creation and the great capabilities demonstrated by diffusion models in image generation, in this work, we extend the use of text-to-image diffusion models for video cartoonization. Concretely, we apply a fine-tuned Stable Diffusion [8] model with an image-to-image setting, conditioned with the image edges (computed with holistically nested edge detection [9]) and CLIP-generated [10]

prompts to translate the keyframes from a source video, ensuring content preservation. To maintain temporal coherence, the keyframes are translated into grids and the style is interpolated with the example-based style propagation algorithm from [11] (see Figure 1 for an example). The experiments on a new video dataset demonstrate the effectiveness of the method and the capability of producing temporally coherent videos while maintaining the semantic information. Our contributions include the following:

- A zero-shot method capable of translating short videos into clips with a specified style.
- A new video dataset of backgrounds without people with different camera movements and various landscapes for reproducibility and benchmarking purposes. It is available in the project repository (https://github.com/gustavorayo/video-to-cartoon (accessed on 26 July 2024)).
- A fine-tuned model with the style of Ryo Takemasa, a Japanese illustrator, that can be used to generate images in his style or integrated with other methods that use text-to-image diffusion models. The model is accessible at Huggingface: https://huggingface.co/gustavorayo/ryo-takemasa-v1 (accessed on 26 July 2024).

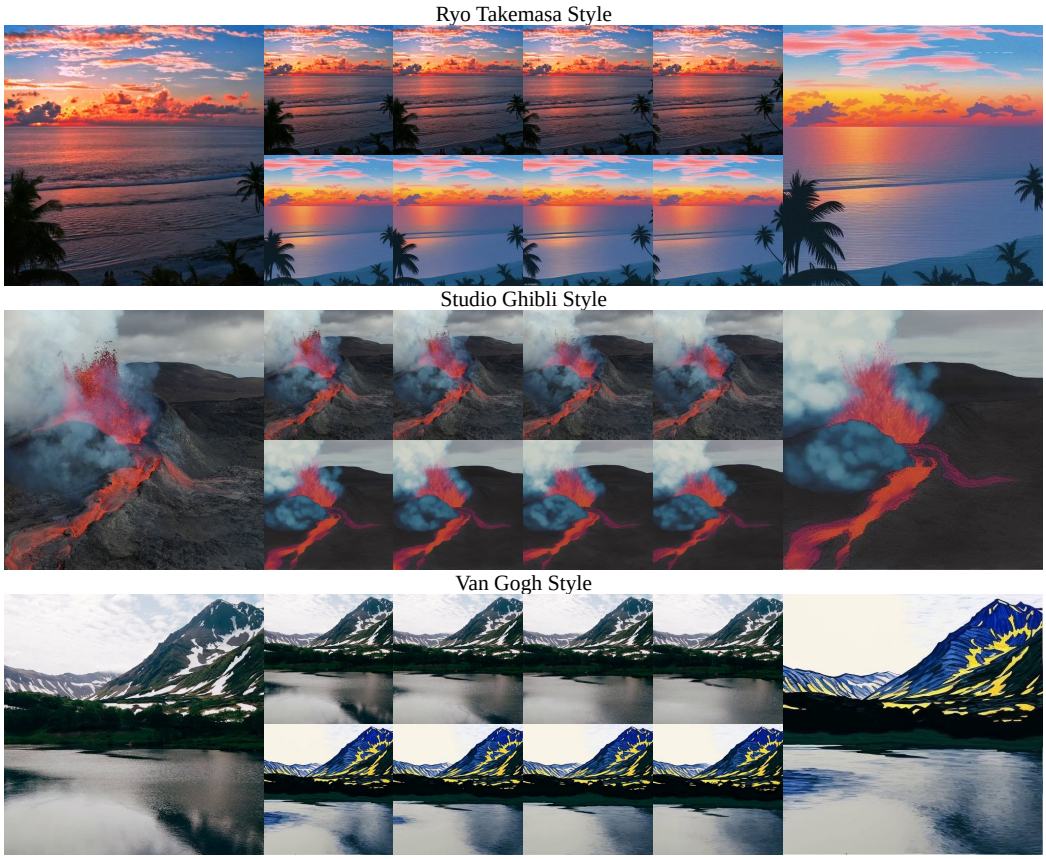


Figure 1. Frames from source videos and translated videos using the proposed method, featuring the three selected styles: Ryo Takemasa, Studio Ghibli, and Van Gogh. The original frames are shown on the left and top of each style, while the corresponding translated frames are displayed on the right and bottom.

2. Related Work

The general objective of video stylization is to apply a (typically artistic) style to a (typically live action) video while preserving its underlying structure and semantics. There are different variants of this problem, leading to different research lines. The style can be derived from a single independent image (style transfer), propagated from some hand-stylized frames (style propagation), specified through text prompts, or learned from a corpus of images. Here, we focus on the latter approach, which intersects with broader research topics like image-to-image translation, video-to-video translation (also known

as translation-based video synthesis), and video editing. The recent success of generative models in image synthesis has established them as the foundational technique for the majority of the methods in the field. The first generative methods to achieve satisfactory results in transforming photos of real-world scenes into cartoon-style images were based on conditional generative adversarial networks, such as CartoonGAN [3], AnimeGAN [12], and the work presented in [13]. Recently, these methods have been outperformed by those based on diffusion models. Pure image-to-image translation methods using diffusion models, like SR3 [14] and Palette [15], operate in a supervised setup, which requires ground truth pairs of images (original and target), making them suitable for tasks like super-resolution or colorization but not for artistic stylization. Therefore, text-to-image diffusion models (e.g., Stable Diffusion [8,16]) are adapted for image-to-image translation using techniques such as SDEdit [17] or ControlNet [18].

2.1. Text-to-Video with Diffusion Models

Most of the video generation methods that use diffusion models are conditioned on text (text-to-video), where a description is provided and the model generates the video based on it. Due to the complexity of training a model to generate videos, many approaches used previously pre-trained text-to-image models with some modifications to enforce temporal consistency.

AnimateDiff [19] and Text2Video-Zero [20] are two recent methods capable of generating videos in some style based on a text description. AnimateDiff [19] extends the personalized text-to-image (T2I) diffusion models based on Stable Diffusion (SD) [8] and converts them into an animation generator. It uses a motion modeling module that is inserted between the pre-trained image layers of a base T2I model and is trained with video clips. During training, the weights of the base T2I models are frozen and only the weights of the module are updated. During inference, the trained module can be used with T2I models fine-tuned with techniques such as DreamBooth or LoRA. This flexibility enables generating animated images with a specific style based on just a text description. Text2Video-Zero [20] exploits the synthesis power of Stable Diffusion (SD) to generate videos. Text2Video-Zero requires as an input a text description and an integer m representing the number of frames. To keep the global scene consistent, the method enriches the latent codes of the generated frames with motion dynamics and implements a cross-frame attention mechanism to preserve the appearance and identity of the foreground object.

While diffusion models have become the de facto approach for generating high-resolution images and videos from natural language inputs, their significant computational costs and long sampling times have driven research into more efficient training methods and faster sampling techniques. One promising path includes Rectified Flow Models [21], a recent generative model formulation. These models create a transport map between two distributions using an ordinary differential equation (ODE), which can be faster to simulate compared to the probability flow ODE in diffusion models. Rectified Flow Models have recently been successfully applied to high-resolution text-to-image synthesis in [16].

Although text-to-video approaches can produce videos with a specific style, they cannot be used in situations requiring specific content and controlled movements. In such cases, it becomes necessary to use a reference video that can be transformed into another video with a new style.

2.2. Video Editing with Diffusion Models

An alternative way to transform a video into a cartoon is by using approaches for video editions. While these models are typically used for making local changes like modifying the attributes of an object, some of them, such as Pix2Video [22], Tune-A-Video [23], and Vid2vid-zero [24], enable global editions including style changes. Consequently, this facilitates the transformation of a video into a cartoon.

Pix2Video [22] takes a sequence of frames of a video clip and generates a new set of images that reflect an edit denoted by a target text. It is built on a Stable Diffusion model

conditioned on depth. Pix2Video injects the features obtained from the first frame (anchor frame) and the previous frame by manipulating the self-attention module of the U-Net [25] network. The previous frame preserves the appearance changes and the anchor frame avoids the forgetting behavior on longer sequences. Additionally, to improve the temporal stability, during the first 25 denoising steps, they use noise correction so that the previous and current frames are as similar as possible.

Tune-A-Video [23] extends and fine-tunes Stable Diffusion with a single video to learn the motions of the video. Then, at inference, it can generate novel videos that represent the edits in text prompts, including changes in the style. The modifications involve changing the 2D convolution layers in the U-Net [25] for pseudo-3D convolution layers and extending the spatial self-attention mechanism to the spatial temporal domain. It uses a sparse version of a causal attention mechanism, where the attention matrix is computed using the latent features of the current frame, previous frame, and first frame. This method has the disadvantage of requiring tuning the model for every video.

Vid2vid-zero [24] requires a sequence of frames and two texts, one with the description of the video and the other with the description of the desired video. Vid2vid-zero first inverses each frame in the input video to obtain the latent noise and null-text embedding and then generates the edited video under cross-attention guidance. For temporal modeling, the pre-trained self-attention in the original U-Net [25] blocks is replaced with cross-frame attention that shares the same weights. Aside from the specific changes in the video, this method can also translate a video into a new style by specifying it in the target prompt.

2.3. Video-to-Video Translation

Alternatives for the direct translation of videos into another style are also available. A recent method is Rerender a video [26], a zero-shot text-guided video-to-video translation that divides the translation process into two parts. In the first part, an adapted diffusion model is used to translate the keyframes into a new style, and then, in the second part, the keyframes are propagated to the rest of the frames.

For keyframe translation, Rerender a video replaces the self-attention layers of the U-Net [25] with cross-frame attention layers. The Key K and Value V of the attention are generated using the first and previous frames. Additionally, to constrain the cross-frame to the local shape and texture consistency, optical flow is used to warp and fuse the latent features in the initial steps. In the middle steps, the previous frames are warped and encoded back to the latent space for fusion. Finally, in the late steps, adaptative instance normalization (AdaIN) is applied to keep the color style coherent throughout all the keyframes. In the second part, a patch-based frame interpolation algorithm is employed that uses color, positional, edge, and temporal guidance for dense correspondence prediction. This model yields good results but comes with a high computational cost, requiring over 24 GB of vRAM.

Another alternative for transforming a video into a cartoon is by using tools such as TemporalKit [27]. TemporalKit serves as an extension for the Stable Diffusion web UI, a browser interface that enables image generation or translation through base Stable Diffusion (SD) or personalized models. This extension adds a new tab to the UI, where users can upload a video and specify the parameters to form a grid image with keyframes. By using the capabilities of Stable Diffusion web UI, users can translate the grid image. The resulting image along with the frames of the video are used by TemporalKit to prepare a folder structure ready to be used by EbSynth Beta, a tool that uses example-based video stylization to propagate the style to the rest of the frames. The disadvantage of TemporalKit is that the process is mostly manual and the translation of the keyframes depends on the knowledge of the user.

3. Methods

Our approach involves a two-stage process. Initially, we select specific keyframes from the source video and translate them into a distinct style during the first stage. The subsequent stage involves propagating the style to the rest of frames using the algorithm

from [11]. The method pipeline is illustrated in Figure 2 and described in more detail in the following subsections.

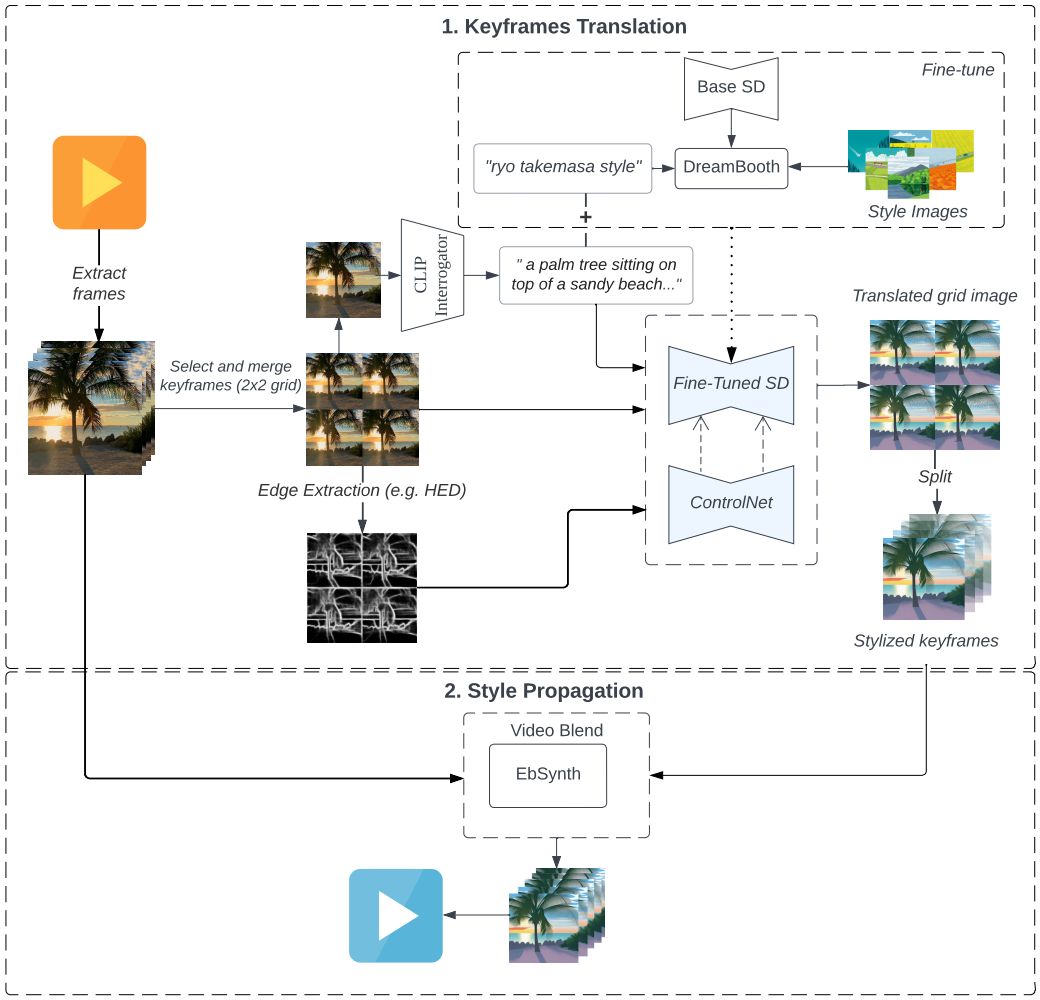


Figure 2. Method pipeline. The process is divided into two main stages: keyframe translation and style propagation executed sequentially as a pipeline. They are represented in dashed boxes as well as their respective subprocesses. In the first stage, four keyframes are selected from the video and combined into a single grid image, which is used for edge extraction. One of the keyframes is used to extract a description using CLIP Interrogator [28]. The resulting description is used to form the final prompt by combining it with the keywords that identify the style in the fine-tuned model (the "+" symbol represents the text concatenation). This prompt, along with the edge map and the grid image, are passed to a subprocess to generate a translated grid image. This subprocess integrates the fine-tuned version of Stable Diffusion (SD) with ControlNet, which conditions the image generation as indicated by the dashed arrows. The resulting translated image is then separated to form the stylized keyframes, which are used in the second stage. In the style propagation stage, the stylized keyframes serve as a reference to propagate the style to the original video frames using EbSynth [11]. The resulting stylized frames are used to create the final cartoonized video. In this figure, solid arrows represent the main flow of data through the process. The dotted arrow indicates that the fine-tuned SD results from the fine-tuning subprocess.

3.1. Keyframe Translation

The keyframe translation requires a fine-tuned Stable Diffusion model with the desired style. The model must be fine-tuned with Dreambooth [29], a fine-tuning approach that implants the style and an identifier into a base Stable Diffusion model. Dreambooth is a

common approach, and we can find fine-tuned models with different styles on platforms like HuggingFace or Civitai.

Once we have a fine-tuned Stable Diffusion model with the desired style, we can proceed with the frame translation. Stable Diffusion enables image generation guided by text (text-to-image) or based on an image (image-to-image). In our case, we want to keep the structure of the frames and change only the style, so we condition the generation on the source frame using image-to-image translation.

The method begins by extracting the frames from the video and translating keyframes into frames with the intended style. A naive approach could be to translate every keyframe using only the identifier of the style in the fine-tuned model. However, this would lead to significant differences in the translated frames. To mitigate this problem and make the process automatic, we employ a more advanced approach. First, we extract the description of one of the keyframes using CLIP Interrogator, a prompt engineering tool that combines CLIP and BLIP [30] to optimize text prompts that match a given image. The identifier and description are then concatenated and used as a prompt in the fine-tuned version of Stable Diffusion.

Translating the frames independently produces noticeable differences between the frames because the model is not aware of the information in the previous and future frames. To mitigate this problem, other methods such as Pix2Video [22] manipulate the self-attention module of the U-Net network to use features from an anchor frame (first) and previous frame. In our case, we follow a different approach. We select four keyframes within an interval and combine those frames in a grid (2×2) to form a single image in the same way as TemporalKit [27]. The interval is calculated by dividing the total number of frames by three and rounding down to the nearest integer. More frames can be used to form the grid, which may be necessary in videos with large motion to achieve better results, but it requires more computing resources. Using a grid will ensure that the resulting frames are consistent and contain the general features present in the video. The grid size dictates how many frames need to be generated by the generative model versus the style propagation algorithm. Due to the constraints of style propagation, the grid size also sets the maximum video length that can be processed. A larger grid enables the processing of longer videos or videos with rapid structural changes that surpass the abilities of style propagation.

For better preservation of the structure, we add additional conditions using ControlNet [18]. ControlNet allows different conditions for controlling the generation or translation of images (e.g., human pose, segmentation maps, and edges). In our case, the videos consist of landscapes without people, so we use conditions based on edges, which effectively captures the shapes and boundaries. Conditions based on human pose would be more appropriate in clips featuring people. Two common alternatives are Canny edges and HED edges. While either alternative could be used, we selected HED edges for our implementation. The edges are extracted from the grid image.

The grid image, the edges, and the prompt are provided to the fine-tuned model, which produces a stylized image with the same dimensions as the original image. The resulting image is then separated to obtain stylized keyframes. These keyframes are then used to propagate the style to the rest of the frames in the next stage.

3.2. Style Propagation

To make the process faster and improve the temporal coherence, our method uses EbSynth [11], an example-based method for video stylization with a focus on preserving the visual quality of the style, reflecting structural changes and maintaining temporal coherence. It does not rely on neural networks. Instead, it uses an implementation of a non-parametric texture synthesis algorithm.

EbSynth could be used to propagate the style to multiple frames using only a single reference keyframe (basic stylization), but it also enables a more advanced stylization requiring some guidance channels (edge, temporal, and positions) for better style propaga-

tion. These guidance channels must be generated separately and provided as arguments. For this reason, we compute the guidance channels following the approach of [26].

The advanced stylization is performed in sequences using a forward pass and a backward pass. A sequence corresponds to the frames between two stylized keyframes. In the forward pass, three guiding channels (G_{temp}, G_{edge}, and G_{pos}) are generated as described in [11] and the style is propagated to the frames in the sequence using the starting keyframe. In the backward pass, different guiding channels are generated and the style is propagated using the keyframe from the end of the sequence. This process produces two separate stylized sequences. Then, each corresponding frame from these sequences is blended to produce the final stylized frames. With 4 keyframes, the process has to be performed three times. The final step is to transform the stylized frames into the new video.

4. Datasets and Style Models

We use two datasets: one composed of images for fine-tuning Stable Diffusion with a custom style and a second dataset consisting of videos to evaluate the results of the method. For fine-tuning Stable Diffusion, we use 100 squared images of 512 pixels containing the Ryo Takemasa style. The video dataset consists of 20 videos extracted from Pexels (https://www.pexels.com/ (accessed on 26 July 2024)). The original dimensions of the videos are 950 × 540 pixels and vary in duration, ranging from 7 s for the shortest to 60 s for the longest. The video dataset is publicly available in the project repository. The videos encompass various camera movements and contain a variety of landscapes, including forests, mountains, roads, buildings, and more. For the experiments, we use only 90 frames center cropped to 512 pixels.

In relation to the style models, we use three distinct ones, each incorporating a different style. We fine-tune one based on the style of Ryo Takemasa, an illustrator based in Nagano, Japan. This model requires the token "ryo takemasa style" in the prompt to generate or translate images in this style. The other two models incorporate the styles of Ghibli Studio and Van Gogh. Both were available at HuggingFace. Ghibli Studio is a Japanese animation studio based in Koganei, Tokio. The model was trained on images from modern anime feature films and needs the token "ghibli style" in the prompts. The model with Van Gogh style was trained with screenshots from the movie Loving Vincent. It requires the token "lvngvncnt" in the prompt.

5. Experimental Setup

For the experiments, we use two environments: one for fine-tuning a Stable Diffusion model with the Ryo Takemasa style and the other for video cartoonization. The first consists of a server equipped with an NVIDIA T4 GPU with 15 GB of vRAM. We employ the Diffusers [31] library to fine-tune stable-diffusion-v1-5 using the DreamBooth technique. We fine-tune the U-Net [25] using 10,000 training steps and a learning rate of 2×10^{-6}. The text encoder is trained for 450 steps with the same learning rate of 2×10^{-6}.

The second environment consists of a a server equipped with an NVIDIA V100 GPU with 16 GB of vRAM. Our code implementation is based on PyTorch [32], and the primary library used is Diffusers [31]. This library contains state-of-the-art diffusion pipelines for generating images, audio, and even 3D structures of molecules. In particular, we use three pipelines: StableDiffusionImg2ImgPipeline for text-guided image-to-image generation, StableDiffusionControlNetPipeline for text-to-image generation using Stable Diffusion with ControlNet guidance, and StableDiffusionControlNetImg2ImgPipeline for image-to-image generation using Stable Diffusion with ControlNet guidance. The final results are generated using StableDiffusionControlNetImg2ImgPipeline (Amazon, Seattle, WA, USA) and the other two for additional experiments.

For automatically extracting the description of a frame, we use CLIP Interrogator [28]. The clip model used was "ViT-L-14/openai" and the blip "blip-large". Huggingface served as the main repository for storing the fine-tuned models. Models with Ghibli and Van Gogh

styles were already available on that platform, and we upload the new model with the Ryo Takemasa style.

6. Metrics

Evaluating a method quantitatively for video cartoonization is a challenging task. A good evaluation should include metrics for style fidelity, content preservation, and temporal coherence. However, there is a lack of metrics to evaluate all these characteristics. Image generative models have been evaluated using Fréchet Inception Distance (FID), a metric that measures the quality and diversity of the generated images. It compares the distribution of generated images with the distribution of real images. This metric has been extended to evaluate diffusion models for videos with Fréchet Video Distance (FVD), which additionally measures the temporal coherence. These metrics have the inconvenience that they do not measure style and content preservation. Furthermore, they require a large sample size, usually greater than 50k [33], making them unfeasible for this work.

The lack of robust metrics has led other works to base the evaluation on user studies [34,35]. For example, Rerender a Video [26], one of the most similar works to our approach, uses a user study with 23 participants to compare the results with other methods. Relying on users has the drawback of leading to subjective human evaluation and might incorporate user biases.

Other approaches [22,23,36,37] incorporate CLIP-based metrics to evaluate some aspects of the models. Following their approach, we use CLIP to assess the temporal coherence and content preservation. The temporal coherence metric consists of computing the cosine similarity between the CLIP embedding of consecutive frames and then computing the average to obtain the final result. The content preservation metric is similar; we use the CLIP embedding of the frames from the source and stylized video and compute the cosine similarity between the embedding of each pair of frames, and then we average the result.

Regarding style fidelity, we encountered challenges in finding feasible metrics. Consequently, we opted to perform a visual analysis and leave this aspect for future work.

7. Results

The method is applied to the 20 videos in the dataset using the three selected styles: Ghibli, Ryo Takemasa, and Van Gogh. The resulting videos are available at https://drive.google.com/file/d/1ErpojLlR_ipJCUxY02rLnpYi1cqSx4ET/view?usp=drive_link (accessed on 26 July 2024). The file contains two folders: one named "1_final_results" with the final results. The second folder, "2_extra", contains two subfolders with additional experiments. One named "naive" employs basic image-to-image translation and basic style propagation. The other is an improvement that uses ControlNet conditions and basic style propagation.

7.1. Style Fidelity and Content Preservation Results

In Figure 3, we can appreciate some frames of two translated videos in each style. As we can see from that figure, the style in the resulting frames is noticeable, and the high perceptual quality is evident. Additionally, the elements are simplified with clearer object boundaries and consistent across the subsequent frames. Another important characteristic is the coherence of the elements across the different frames. In some cases, the shapes are transformed to match the style, as we can see with the trees in the first row (Ryo Takemasa style), but still recognizable for a human. Despite the simplification, subtle elements such as object shadows or light reflection in the water are preserved, as observed in the second row (Ryo Takemasa style).

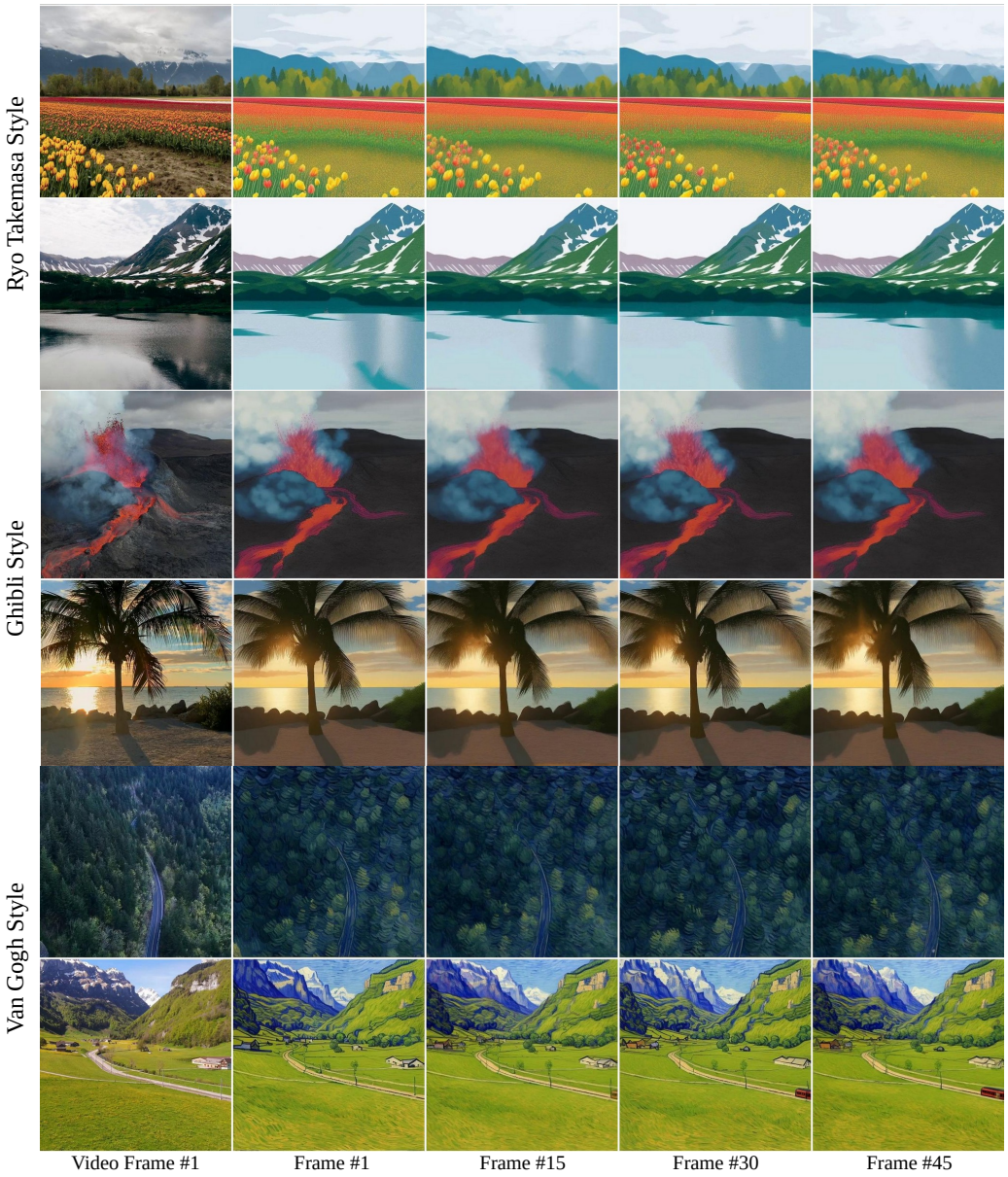


Figure 3. Qualitative results. In the first column, we show the first frame from four videos. The other four frames were extracted from the corresponding translated videos.

The simplification can be also observed in the videos using the Ghibli style. For example, in the frames of the third row, the shape of the smoke has been transformed into a shape similar to a drawing. The texture of the ground from the frames in the third and fourth rows is constant, only changing between the element boundaries. The results for the Van Gogh style (last two rows) also show good style fidelity and that the content of the video is preserved and recognizable even in cases where the elements are transformed to match the style, as in the forest of the fifth row. It is worth noting that the method produces good results even in cases where certain elements are not present in all the frames. For example, in the last row, the train is initially absent in the first frames but then appears in the following frames, as in the original video.

In addition to the visual results, we measure the content preservation using CLIP-based metrics, following related works [22,23,36]. We use the CLIP embedding of the frames from the source and stylized video and compute the cosine similarity between the embedding of each pair of frames, and then we average the result.

The quantitative results for the content preservation are summarized in Table 1. The results show varying values between the videos. In the Ryo Takamasa style, for example, the video with the highest value has a 0.94 average CLIP cosine similarity between the original frames and the translated frames, while the lowest value is 0.69. Figure 4 shows the frames for these two cases. In the worst case, we can see that the shape of the trees is simplified and the color is changed to match the style. However, the general content is still recognizable. The difference in content preservation is present in the other two styles as well.

Table 1. Content preservation results. Average CLIP cosine similarity between original frames and translated frames. The bold numbers are the highest values, and the underlined numbers are the lowest values for each column. In the last column, we show the total average for each style next to the overall minimum value.

Video Identifier	Ryo Takemasa	Ghibli	Van Gogh
01	0.8586	0.8993	0.9401
02	<u>0.6905</u>	0.8118	0.7492
03	0.8852	**0.9510**	0.9221
04	0.7673	0.8334	0.7602
05	0.7434	0.8233	0.8754
06	0.8839	0.8494	0.8884
07	0.8063	0.8639	0.8090
08	0.7845	0.8815	0.9083
09	0.7806	<u>0.7731</u>	0.7688
10	0.8508	0.8440	0.8667
11	0.8112	0.8262	0.8408
12	0.8467	0.8976	0.7792
13	0.7314	0.8182	0.7511
14	0.9118	0.9200	0.8990
15	0.7733	0.8441	0.8553
16	**0.9367**	0.9288	0.8986
17	0.8027	0.8016	0.8994
18	0.8835	0.9215	<u>0.7072</u>
19	0.8235	0.8568	0.8022
20	0.8573	0.9366	**0.9416**
Avg.	0.8215	0.8641	0.8431

Figure 4. Content preservation. Best and worst cases for Ryo Takemasa style.

7.2. Temporal Coherence Results

Temporal coherence is a fundamental property in videos, and we aim to produce videos that have this property. The temporal coherence metric consists of computing the cosine similarity between the CLIP embedding of consecutive frames and then computing the average to obtain the final result. The results are reported for all the videos in Table 2. The results reveal excellent temporal coherence in the resulting videos. The overall average CLIP cosine similarity for all the styles is 0.99. As a reference, [22] reported 0.976 average temporal coherence using the same metric. The high level of coherence is consistent

across every video of the dataset in all the styles, with minimum average values no lower than 0.98.

Table 2. Temporal coherence. Average CLIP cosine similarity between embedding of consecutive frames for 20 videos in the three styles. The underlined numbers are the lowest values for each column.

Video Identifier	Ryo Takemasa	Ghibli	Van Gogh
01	0.9949	0.9963	0.9982
02	0.9928	0.9947	0.9957
03	0.9912	0.9904	0.9885
04	0.9930	0.9947	0.9952
05	0.9919	0.9943	0.9956
06	0.9876	0.9872	0.9868
07	0.9862	0.9841	0.9859
08	0.9922	0.9954	0.9962
09	0.9925	0.9930	0.9979
10	0.9949	0.9918	0.9917
11	0.9968	0.9973	0.9984
12	0.9953	0.9954	0.9957
13	0.9940	0.9942	0.9935
14	0.9923	0.9945	0.9948
15	0.9950	0.9905	0.9968
16	0.9975	0.9972	0.9982
17	0.9979	0.9947	0.9963
18	0.9960	0.9961	0.9956
19	0.9976	0.9971	0.9945
20	0.9929	0.9952	0.9965
Avg.	0.9936	0.9937	0.9946

7.3. Ablation Study

When conducting image-to-image translation with Stable Diffusion, one of the most important parameters is the denoising strength. It controls how much Gaussian noise is added to the reference image before proceeding with the synthesis process. It is important to find the right amount of noise that enables stylization without destroying the image (Appendix A includes examples of image-to-image translation with different denoising strength). The denoising strength can be increased as more conditions are added. However, the fidelity of the style can be affected by the conditions. In Figure 5, we can see the effects and how using a description and ControlNet improves the preservation of the content of the original image (Appendices B–F include examples of image-to-image translation with different conditions and parameterizations).

These conditions also have a great impact on temporal coherence. In Figure 6, we show how these conditions increase the similarity in consecutive frames. For simplicity and better representation, in this graph, we use a basic stylization, propagating the style using only the reference keyframe and without additional guiding channels. In Figure 7, we show the temporal coherence results using a basic stylization versus the advanced stylization, where the style is propagated in sequences based on two keyframes and using additional guiding channels. In the advanced stylization, we can notice a great improvement in the temporal coherence, removing the differences between the frames based on different keyframes, even though there is a small reduction in the similarity between the frames based on the same keyframes.

Generating frames independently has the drawback that the model is not aware of the content of previous and future frames, generating different details in each frame and losing consistency. To address this issue, a more effective approach involves merging all the source keyframes into a single image, translating the combined image, and then splitting it back into individual keyframes. Figure 8 shows the CLIP cosine similarity

between consecutive frames using this method compared to the similarity achieved when generating the frames independently.

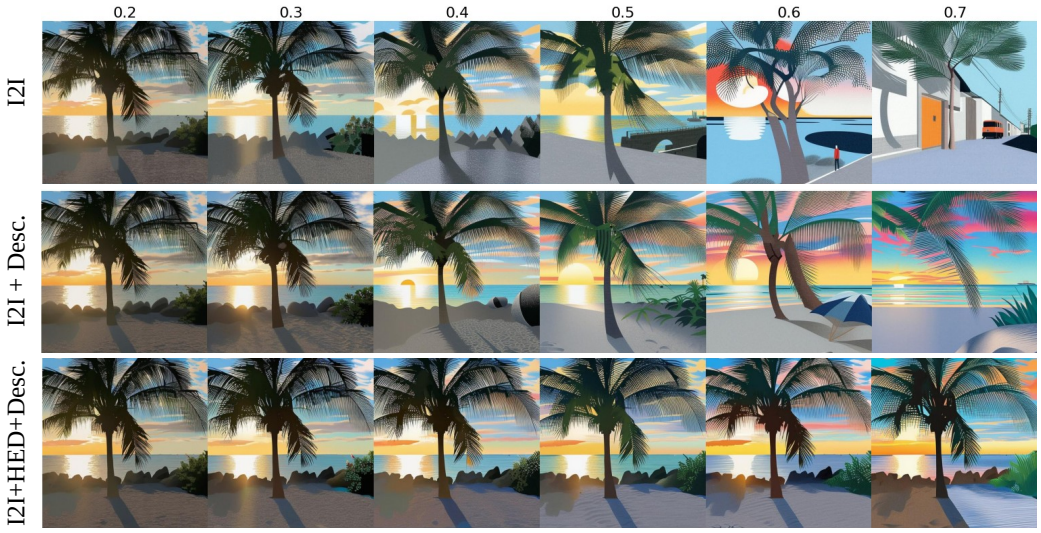


Figure 5. Ablation study: We present variations in translated images under different conditions with progressively increasing denoising strengths. The initial row illustrates outcomes achieved through image-to-image translation using only the style identifier. The second row shows the results when including a description extracted automatically, while the third row showcases outcomes with the additional inclusion of edge conditions using ControlNet.

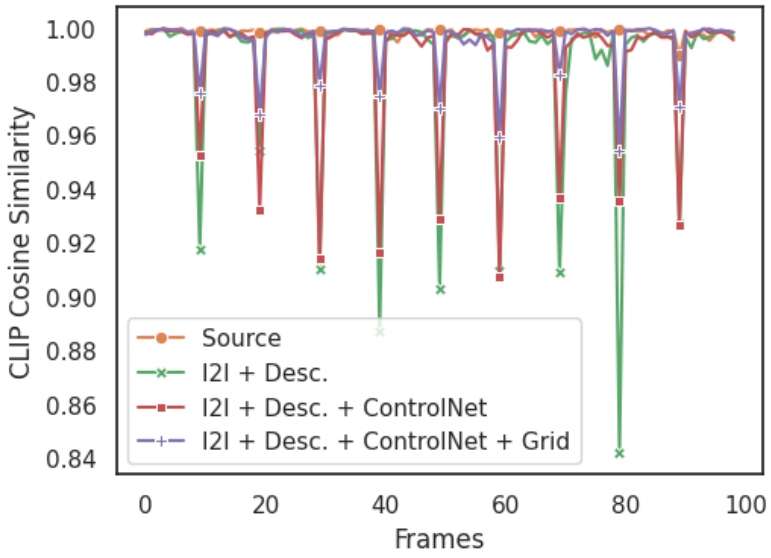


Figure 6. Temporal coherence for a video translated with different conditions. The temporal coherence improves as more conditions are added, especially when translating the frames using grid images.

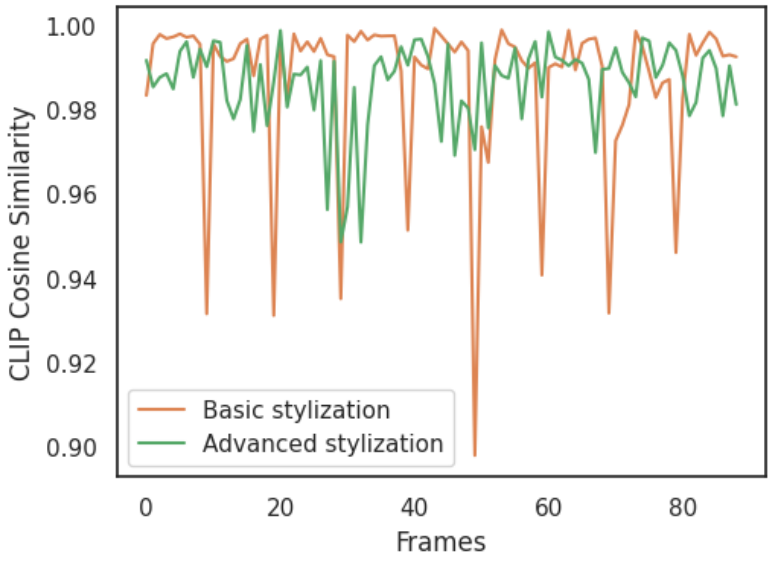


Figure 7. Temporal coherence for basic and advanced stylization. The basic stylization produces clear differences in the stylized frames based on different keyframes, while the advanced stylization improves the temporal coherence.

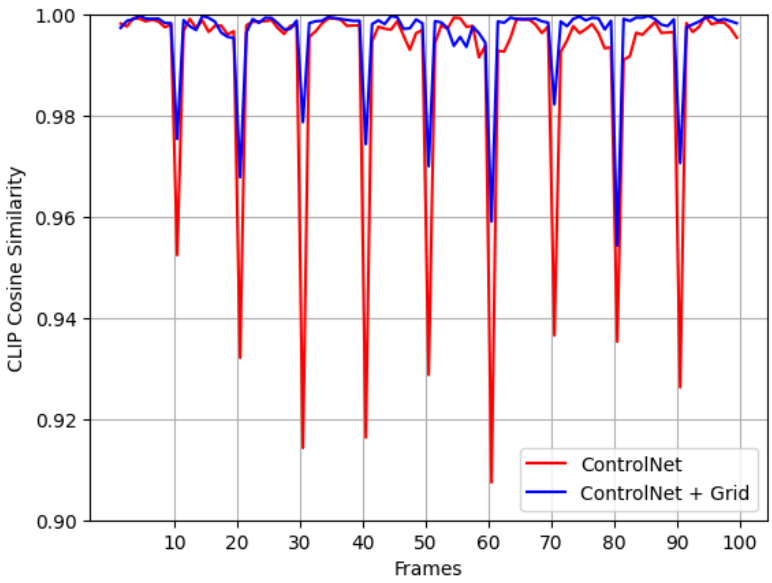


Figure 8. Temporal coherence difference between generating frames independently (red line) vs. generating them using grid images (blue line).

8. Conclusions

This research aimed to develop and apply diffusion models, alongside supplementary tools and techniques, to transform short videos of backgrounds or landscapes into temporally coherent cartoons. This goal was achieved through the creation of an innovative method integrating multiple algorithms, primarily [8,11], enabling the automatic stylization of the short videos or video segments.

A new video dataset has been created and made public for reproducibility and benchmarking purposes. The quantitative and qualitative evaluation demonstrated great capabilities for translating short videos using only the reference video and the specified style, resulting in outstanding temporal coherence. The ablation study underscored the importance of employing multiple conditions and the use of the image grid to enhance the coherence and content preservation. However, it is worth noting that the inclusion of more

conditions comes at the expense of reducing the fidelity of the style. It should be noted that all the videos underwent translation using a uniform configuration, implying that superior results could be achieved through customized configurations for individual videos. Furthermore, the introduction of a new dataset and a new fine-tuned model with the Ryo Takemasa style enabled us to evaluate the viability and effectiveness of the proposed method with a custom artistic style that differs from those used during the training of the base generative model. Despite the visual abstraction of the style and the limited dataset of only 100 images, the method demonstrates solid style fidelity, structure preservation, and temporal consistency, highlighting its promising potential for applying custom styles.

Author Contributions: Conceptualization, G.R. and R.T.; methodology, G.R. and R.T.; software, G.R.; resources, G.R. and R.T.; data curation, G.R. and R.T.; writing—original draft preparation, G.R. and R.T.; writing—review and editing, G.R. and R.T.; supervision, R.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Spanish Ministry of Science and Innovation under contract PID2019-107255GB, and by the SGR programme 2021-SGR-00478 of the Catalan Government.

Data Availability Statement: The dataset and the code are publicly available at https://github.com/gustavorayo/video-to-cartoon (accessed on 26 July 2024).

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

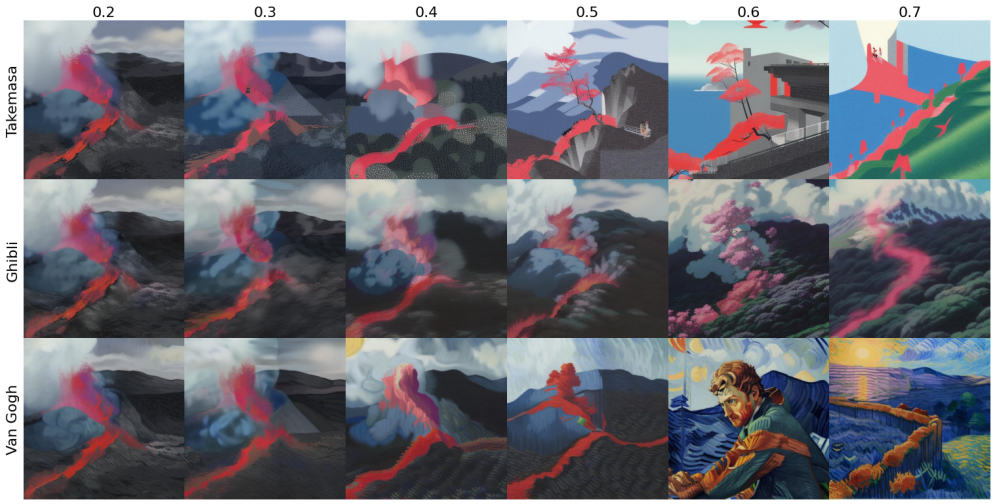
Abbreviations

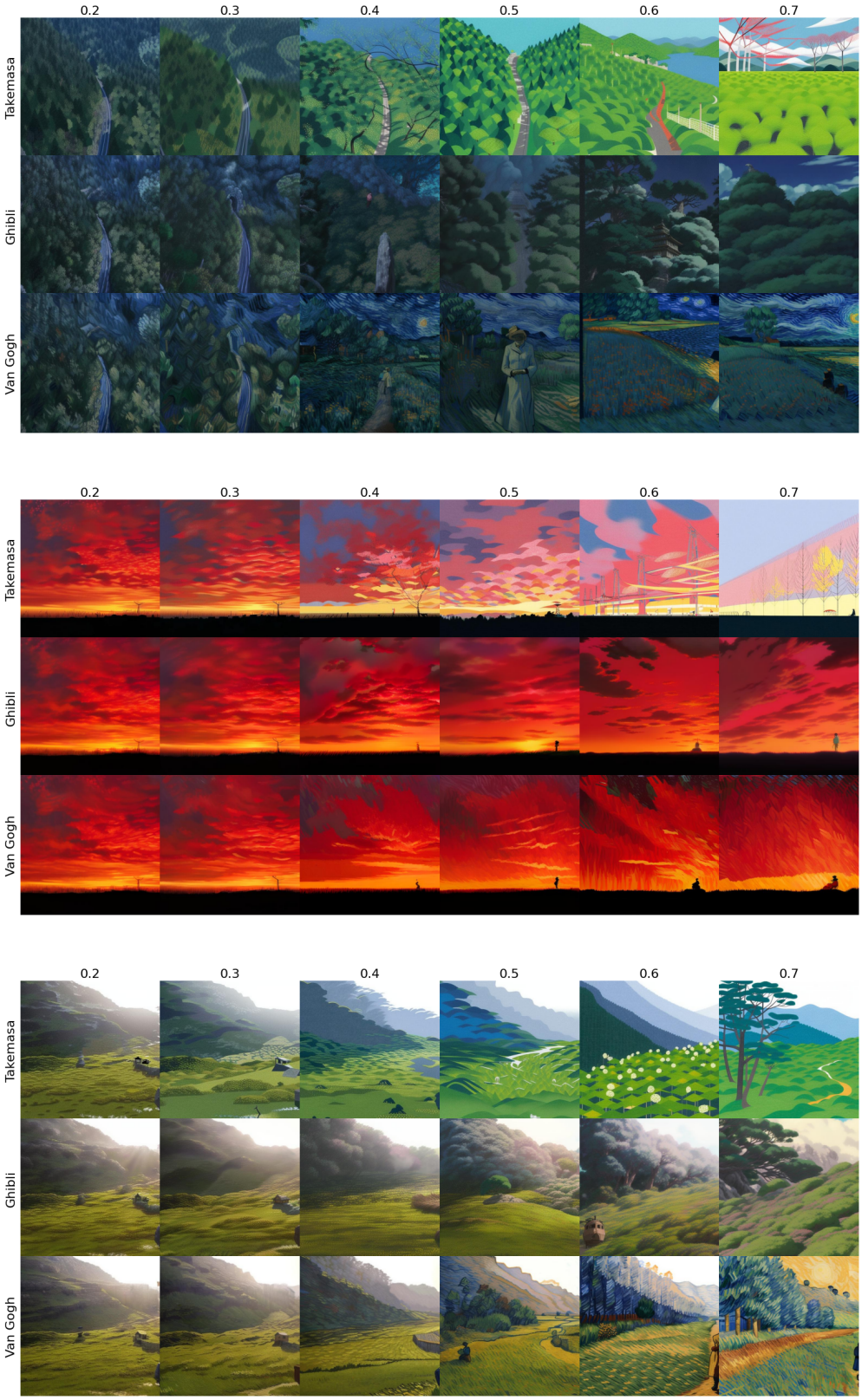
The following abbreviations are used in this manuscript:

BLIP	Bootstrapping Language–Image Pre-training
CLIP	Contrastive Language–Image Pre-Training
GAN	Generative Adversarial Networks
HED	Holistically Nested Edge Detection
SD	Stable Diffusion
T2I	Text-to-Image

Appendix A. Image-to-Image Translation

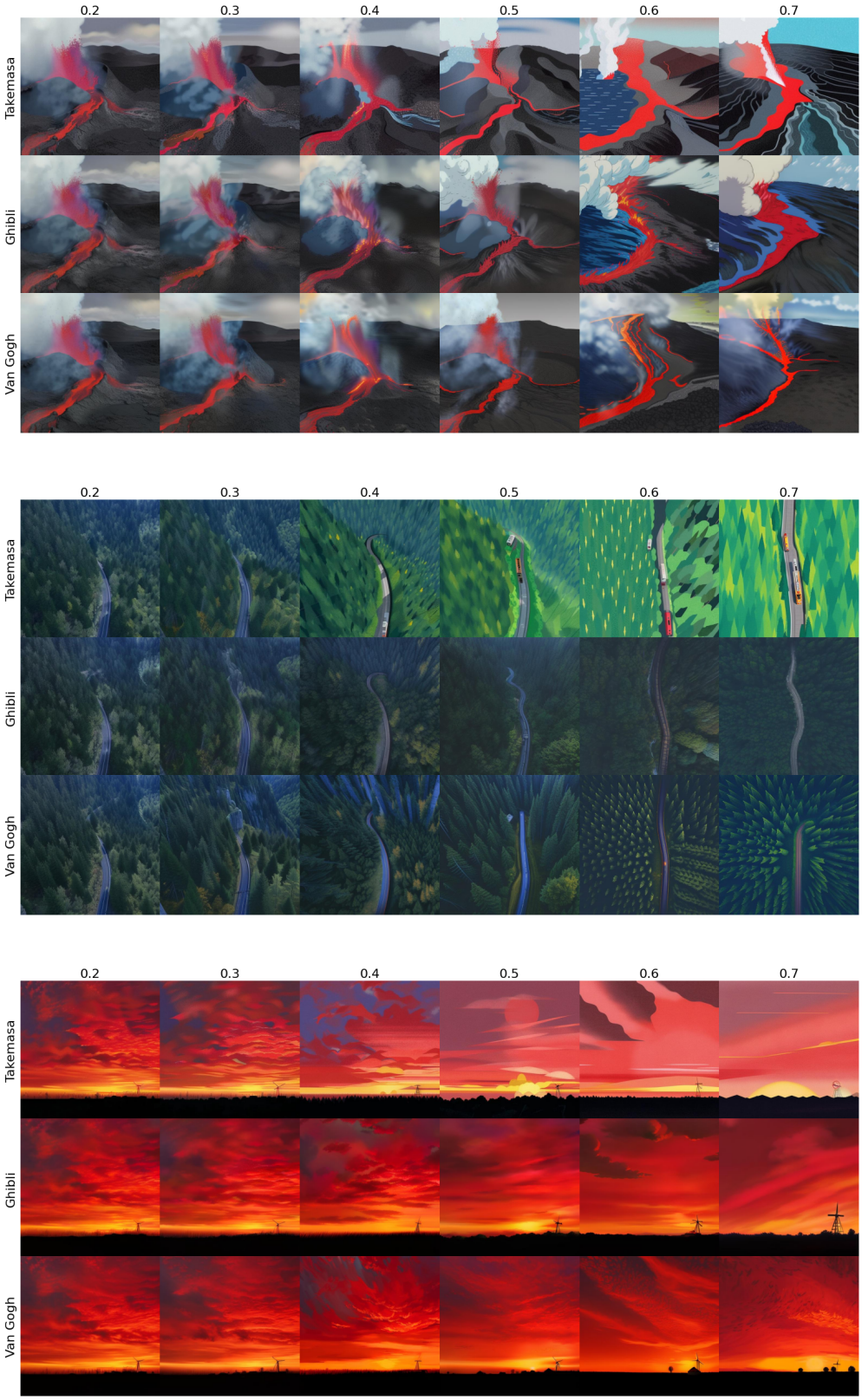
Examples of image-to-image translation to three styles with increasing denoising strength. Parameters: prompt= style keywords, denoising strength= from 0.2 to 0.7, inference steps = 25, and guidance scale = 7.

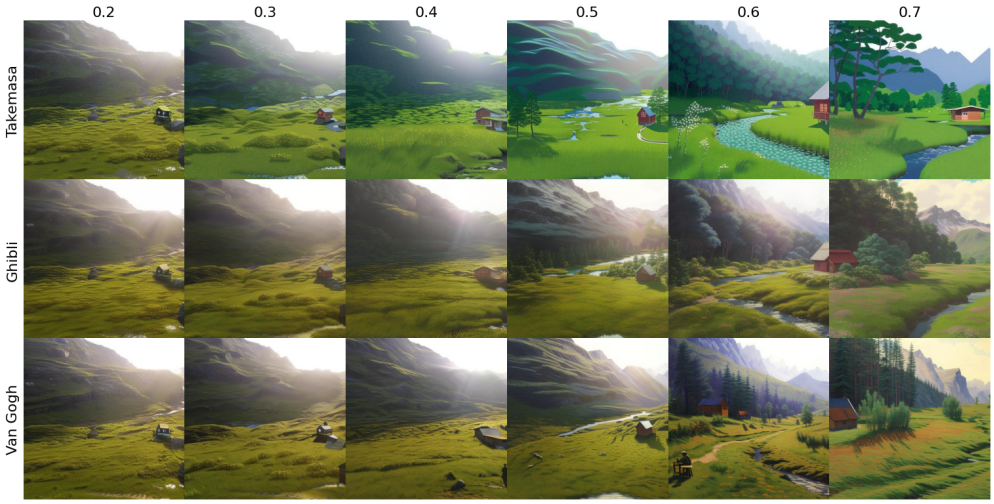




Appendix B. Image-to-Image Translation with Content Description

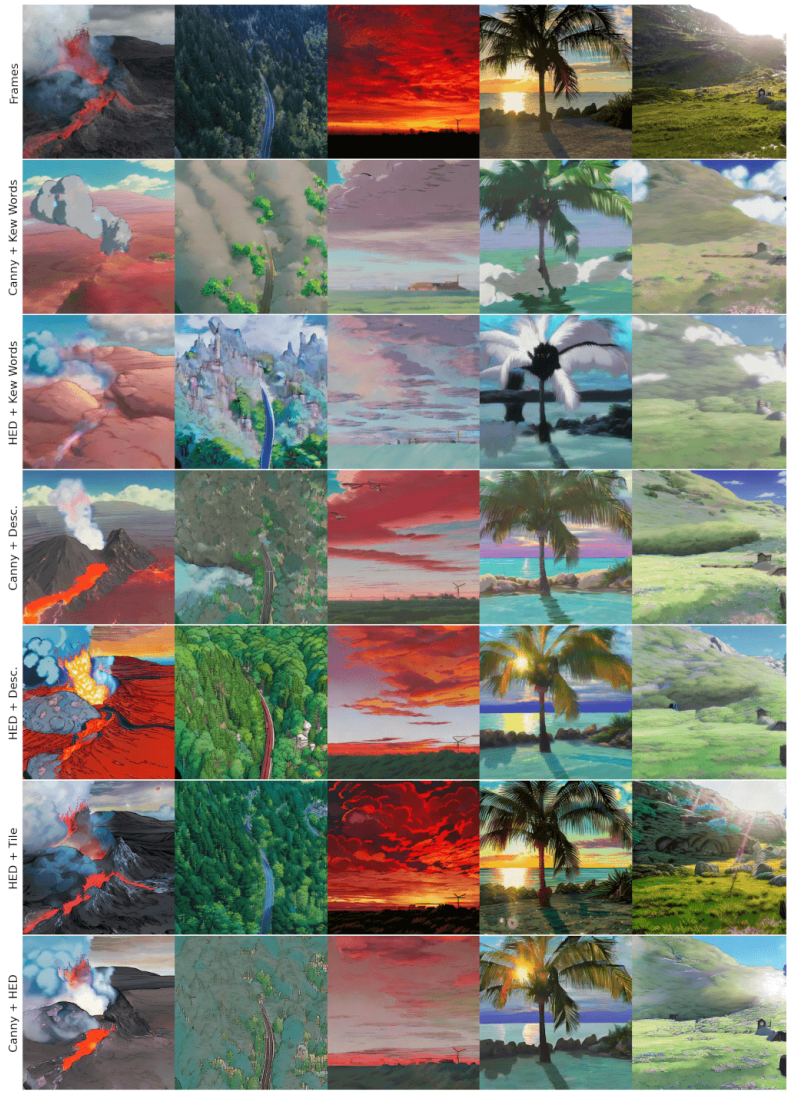
Examples of image-to-image translation to three styles with increasing denoising strength and a description of the image content. Parameters: prompt = style keywords + description extracted using CLIP Interrogator, denoising strength = from 0.2 to 0.7, inference steps = 25, and guidance scale = 7.

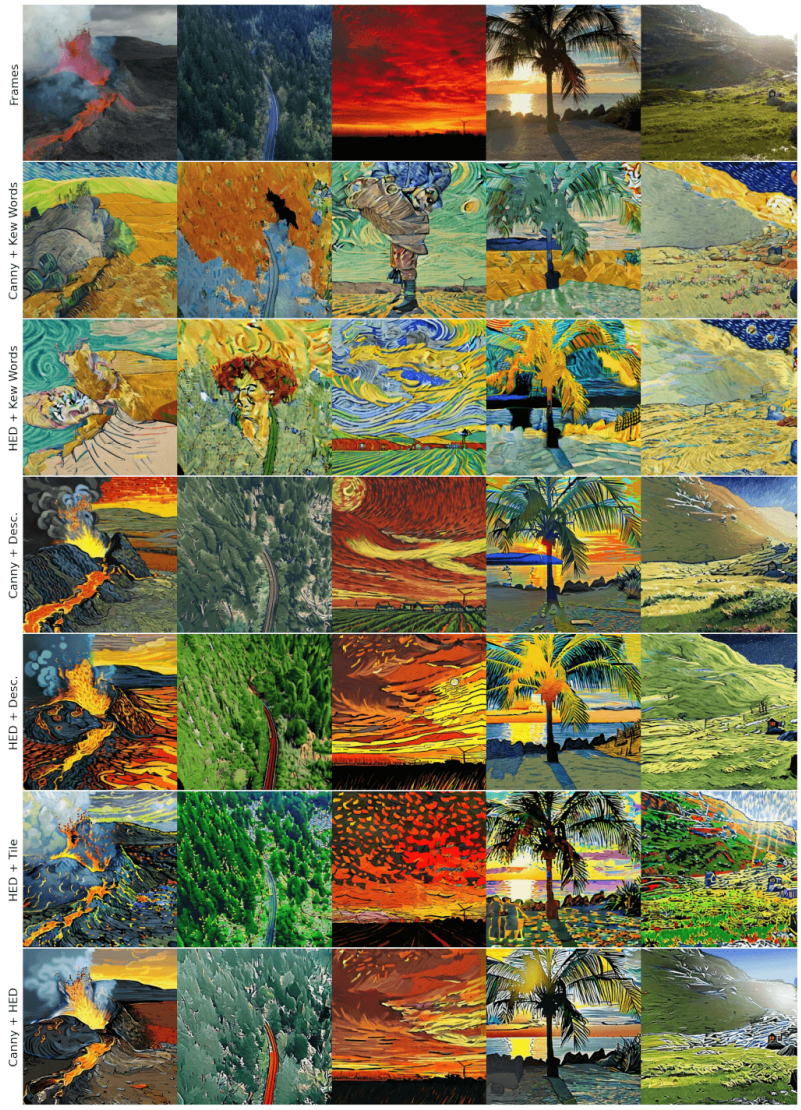

0.2
0.3
0.4
0.5
0.6
0.7
Takemasa
Ghibli
Van Gogh
0.2
0.3
0.4
0.5
0.6
0.7
Takemasa
Ghibli
Van Gogh
0.2
0.3
0.4
0.5
0.6
0.7
Takemasa
Ghibli
Van Gogh



Appendix C. Text-to-Image with ControlNet Conditions

Examples of image generation guided by text and ControlNet conditions using the following parameters: prompt = style keywords or keywords plus description extracted using CLIP Interrogator, inference steps = 25, and guidance scale = 7.5.

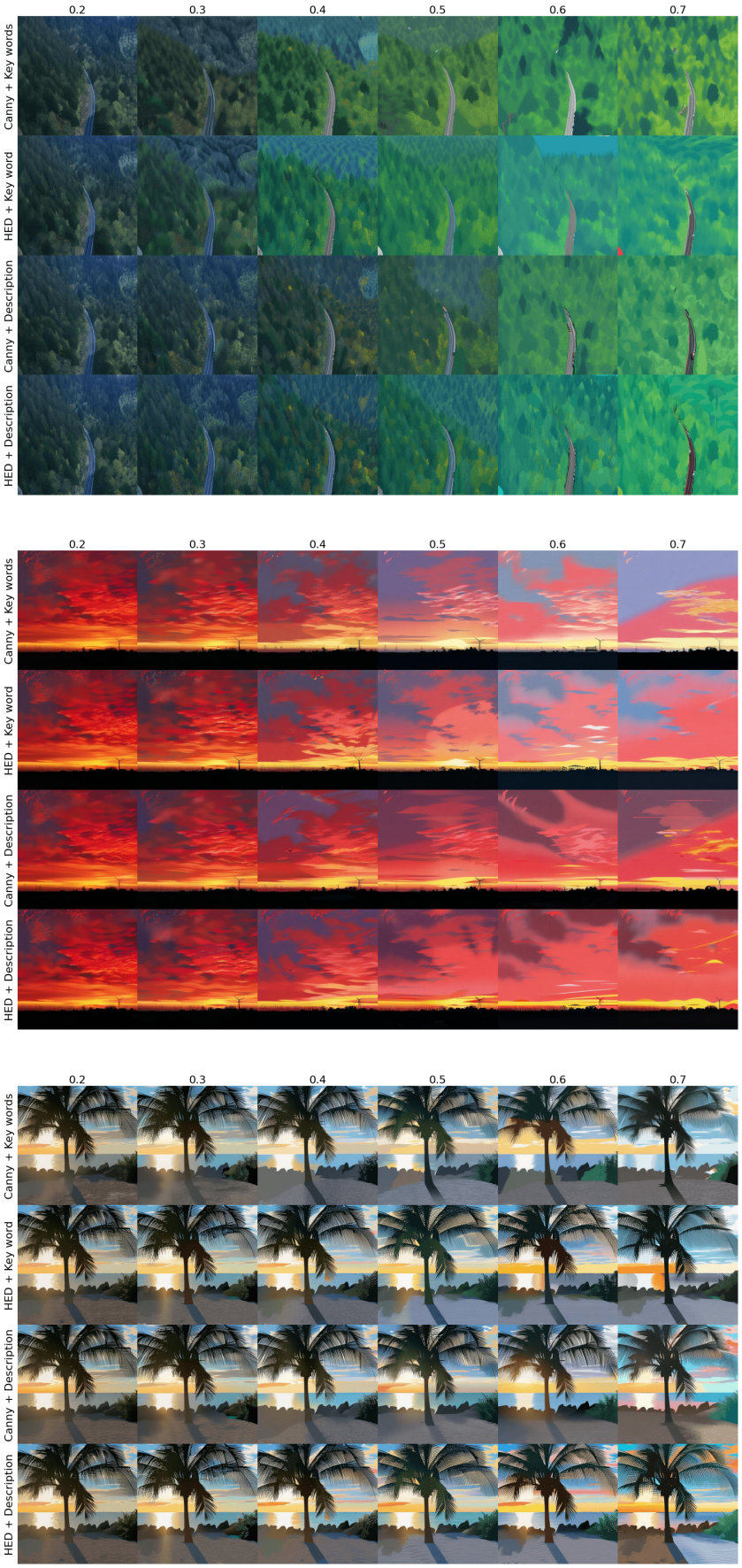


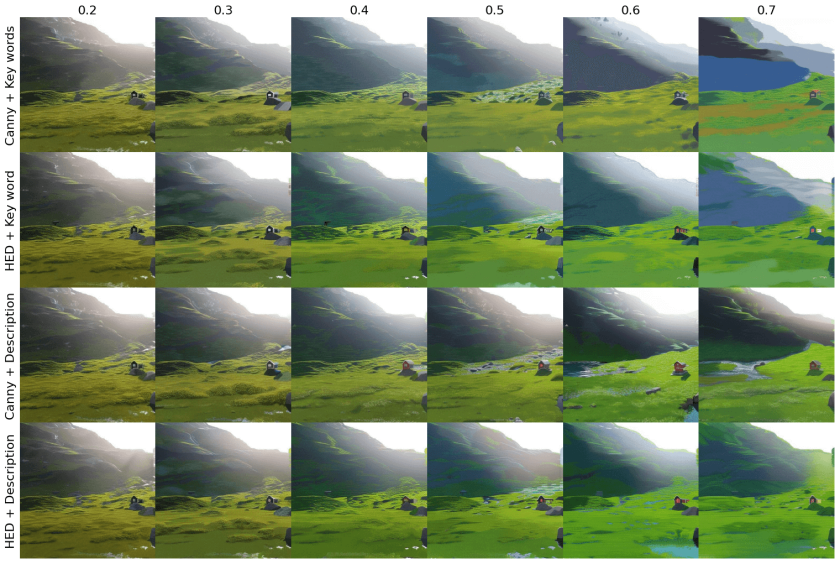


Appendix D. Image-to-Image Translation with ControlNet Conditions

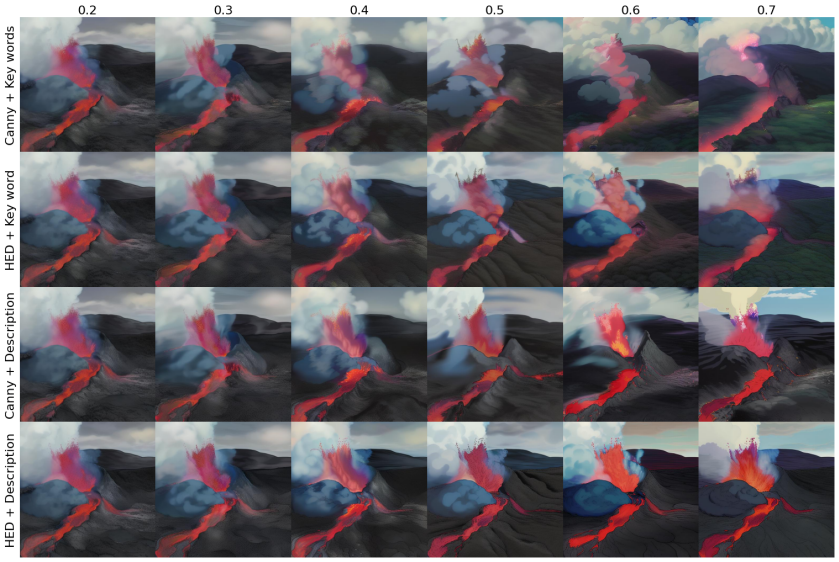
Examples of image-to-image translation with various conditioning parameters for three different styles. Parameters: prompt = style keywords or keywords plus description extracted using CLIP Interrogator, denoising strength = from 0.2 to 0.7, ControlNet conditions = HED or Canny edges, inference steps = 25, and guidance scale = 7.5.

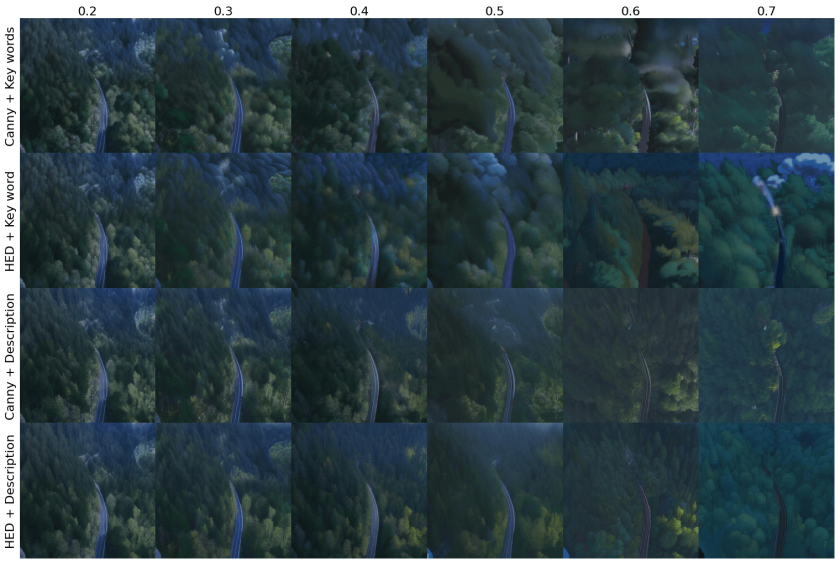
Appendix D.1. Ryo Takemasa Style

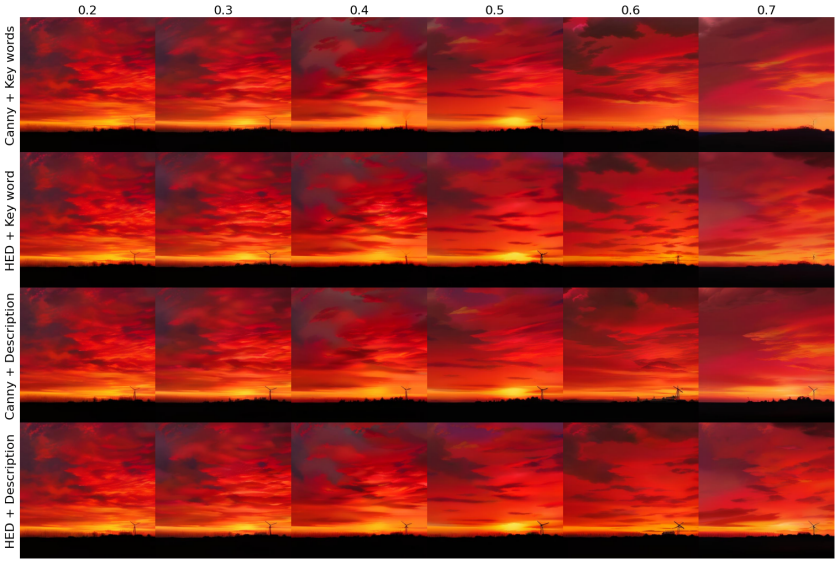




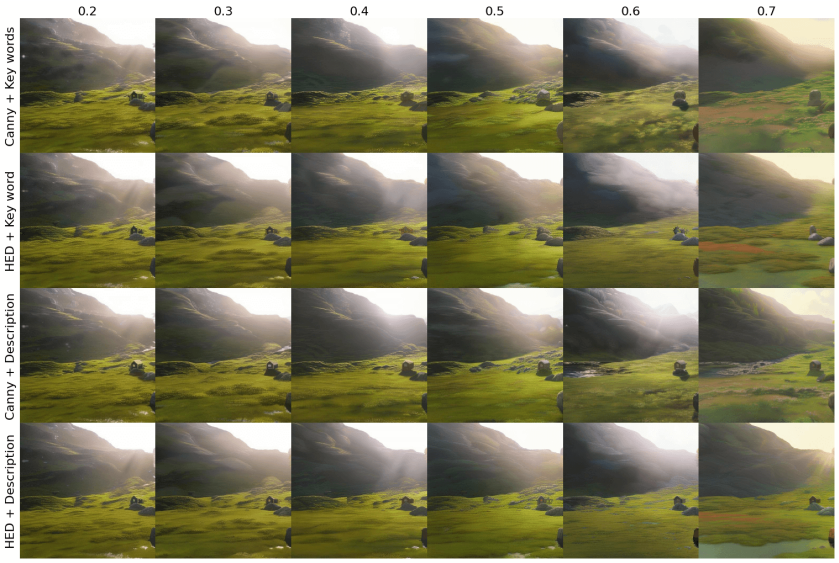
Appendix D.2. Ghibli Style



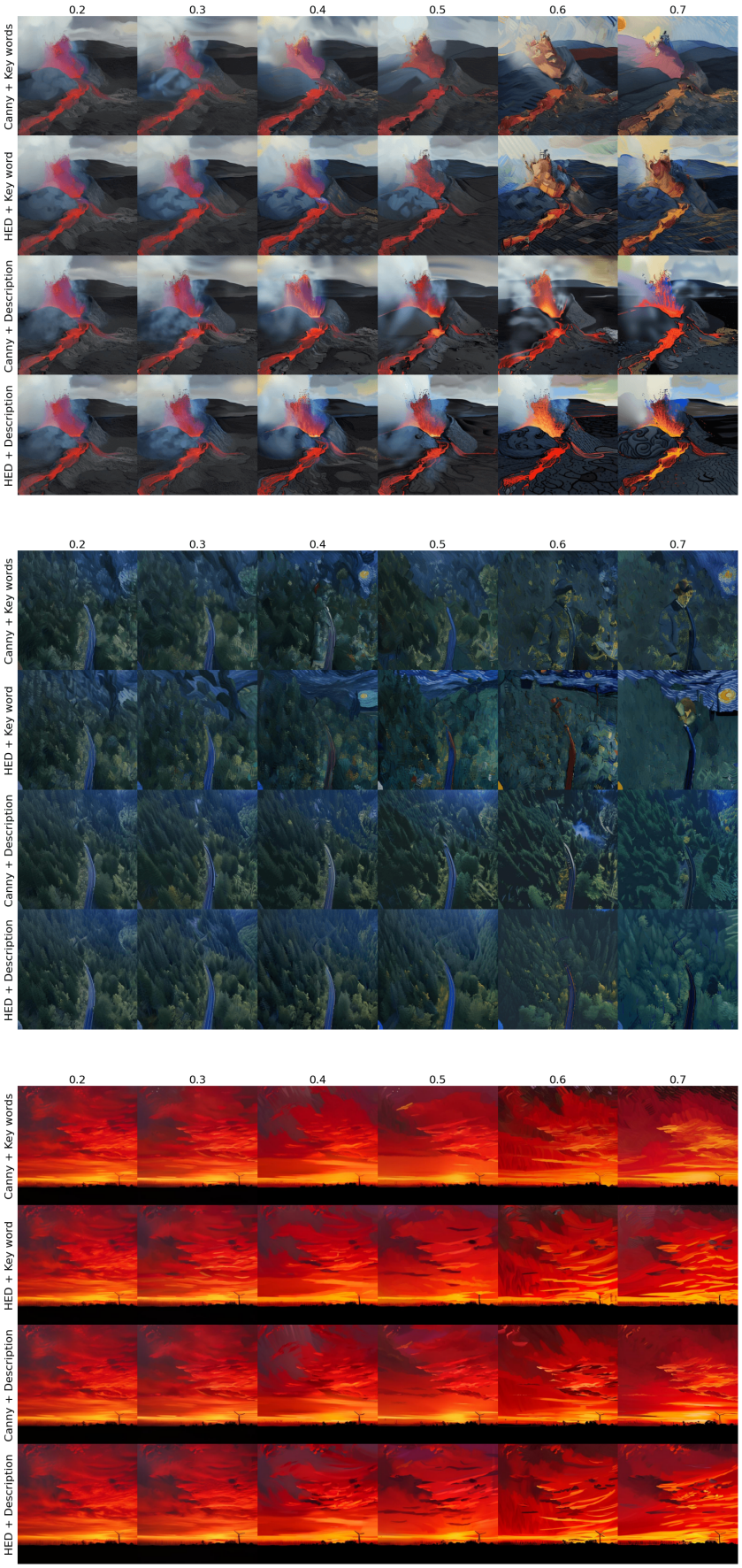


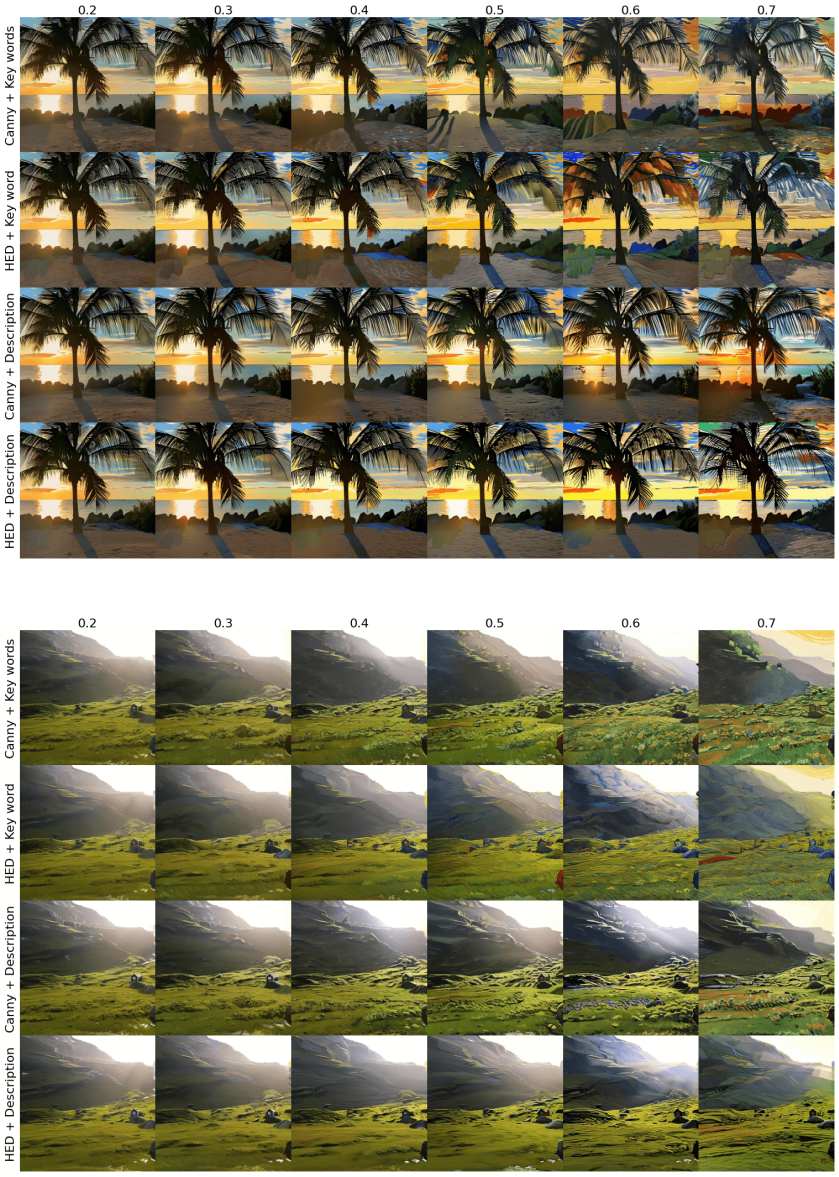

0.2
0.3
0.4
0.5
0.6
0.7
Canny + Key words
HED + Key word
Canny + Description
HED + Description


0.2
0.3
0.4
0.5
0.6
0.7
Canny + Key words
HED + Key word
Canny + Description
HED + Description


0.2
0.3
0.4
0.5
0.6
0.7
Canny + Key words
HED + Key word
Canny + Description
HED + Description

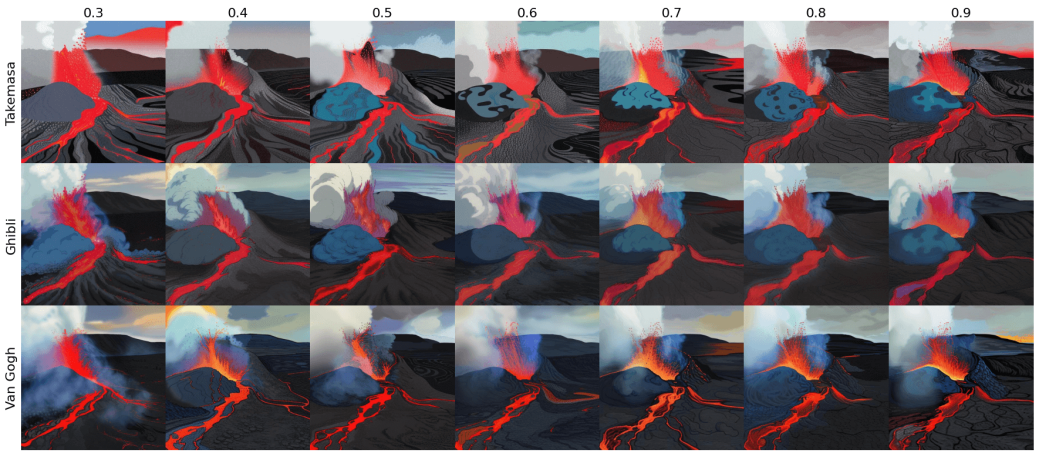
Appendix D.3. Van Gogh Style

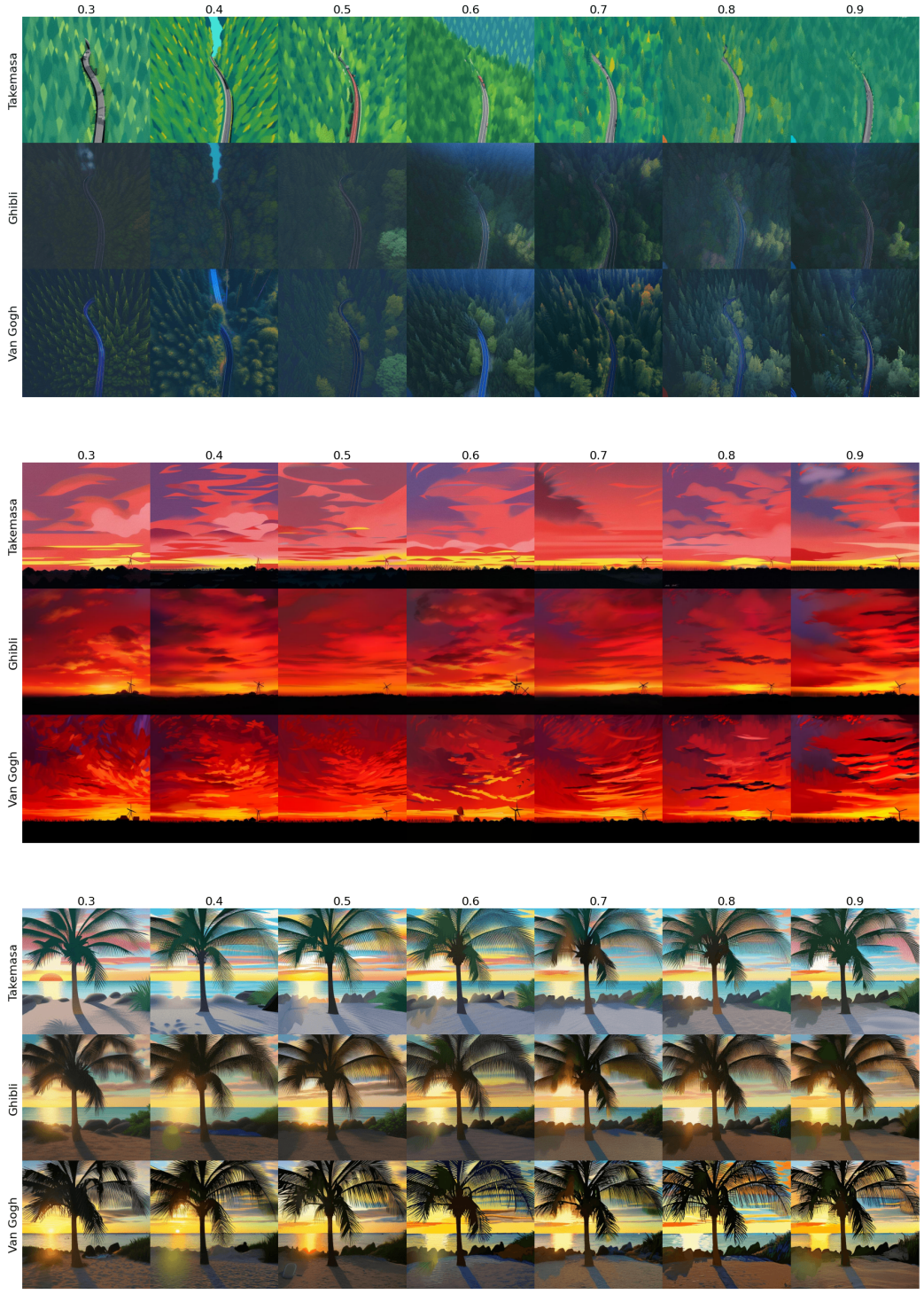




Appendix E. Image-to-Image Translation with Conditions on Weight

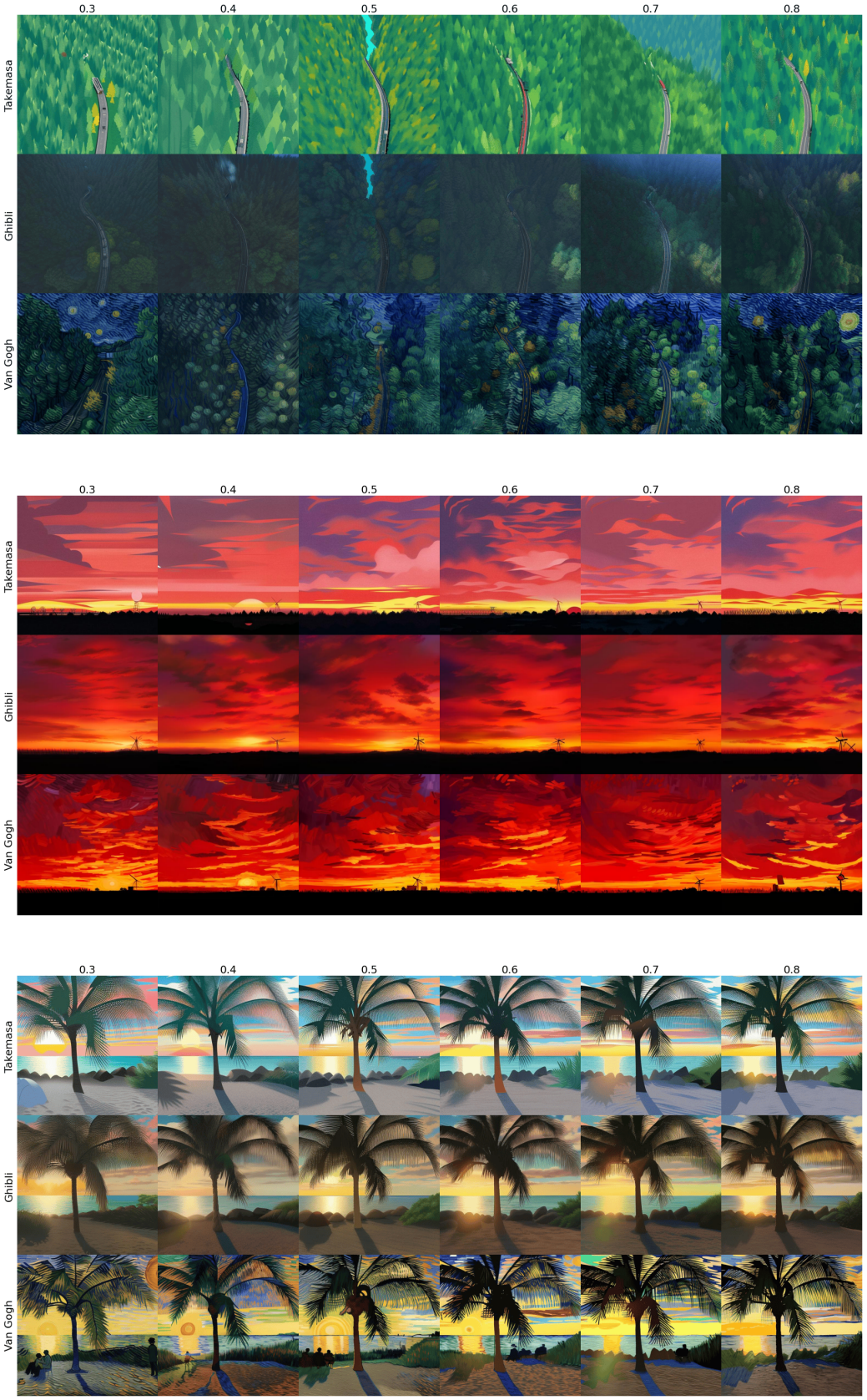
Parameters: prompt = style keywords or keywords plus description extracted using CLIP Interrogator, denoising strength = 0.6, ControlNet conditions = HED or Canny edges, inference steps = 25, guidance scale = 7, and condition weight = from 0.3 to 0.9.

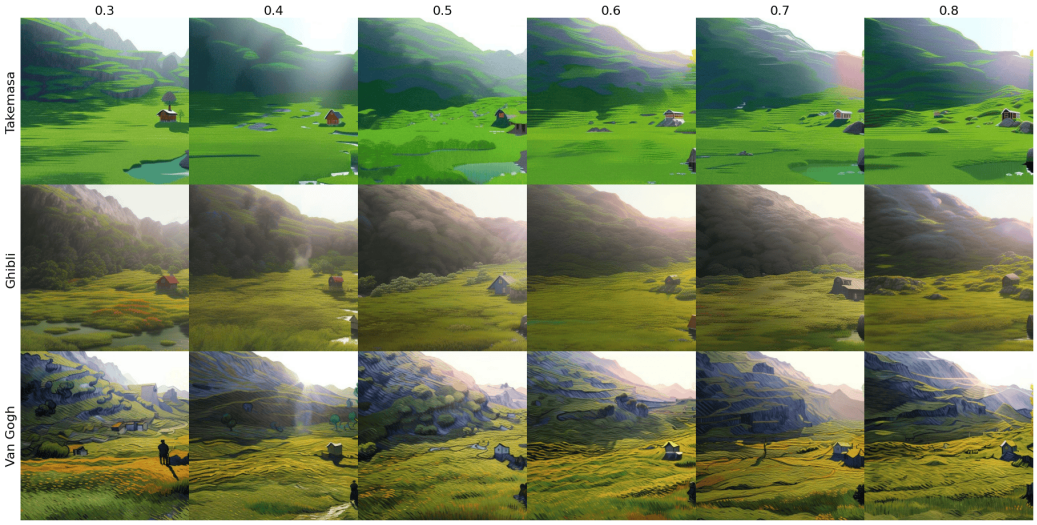




Appendix F. Image-to-Image Translation with Conditions on Limited Steps

Examples of image-to-image translation with ControlNet conditions applied on increasing percentages of denoising steps. Parameters: prompt = style keywords plus description extracted using CLIP Interrogator, denoising strength = 0.6, ControlNet conditions = HED or Canny edges, inference steps = 25, guidance scale = 7, condition weight = 0.7, and condition guidance end = from 0.3 to 0.8.


0.3
0.4
0.5
0.6
0.7
0.8
Takemasa
Ghibli
Van Gogh
0.3
0.4
0.5
0.6
0.7
0.8
Takemasa
Ghibli
Van Gogh
0.3
0.4
0.5
0.6
0.7
0.8
Takemasa
Ghibli
Van Gogh



References

1. 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.
2. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International conference on COMPUTER Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
3. Chen, Y.; Lai, Y.K.; Liu, Y.J. Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9465–9474.
4. Chen, Y.; Pan, Y.; Yao, T.; Tian, X.; Mei, T. Mocycle-gan: Unpaired video-to-video translation. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 647–655.
5. Yang, S.; Jiang, L.; Liu, Z.; Loy, C.C. Vtoonify: Controllable high-resolution portrait video style transfer. *ACM Trans. Graph. (TOG)* **2022**, *41*, 203. [CrossRef]
6. Dhariwal, P.; Nichol, A. Diffusion models beat gans on image synthesis. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 8780–8794.
7. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
8. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. 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; pp. 10684–10695.
9. Xie, S.; Tu, Z. Holistically-Nested Edge Detection. *Int. J. Comput. Vision* **2017**, *125*, 3–18. [CrossRef]
10. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Virtual Event, 18–24 July 2021; Volume 139, pp. 8748–8763.
11. Jamriška, O.; Sochorová, Š.; Texler, O.; Lukáč, M.; Fišer, J.; Lu, J.; Shechtman, E.; Sỳkora, D. Stylizing video by example. *ACM Trans. Graph. (TOG)* **2019**, *38*, 107. [CrossRef]
12. Chen, J.; Liu, G.; Chen, X. AnimeGAN: A Novel Lightweight GAN for Photo Animation. In *Artificial Intelligence Algorithms and Applications*; Springer: Singapore, 2020; pp. 242–256. [CrossRef]
13. Liu, Z.; Li, L.; Jiang, H.; Jin, X.; Tu, D.; Wang, S.; Zha, Z. Unsupervised Coherent Video Cartoonization with Perceptual Motion Consistency. *arXiv* **2022**, arXiv:2204.00795. [CrossRef]
14. Saharia, C.; Ho, J.; Chan, W.; Salimans, T.; Fleet, D.J.; Norouzi, M. Image super-resolution via iterative refinement. *arXiv* **2021**, arXiv:2104.07636.
15. Saharia, C.; Chan, W.; Chang, H.; Lee, C.; Ho, J.; Salimans, T.; Fleet, D.; Norouzi, M. Palette: Image-to-Image Diffusion Models. In Proceedings of the ACM SIGGRAPH 2022 Conference Proceedings, New York, NY, USA, 7–11 August 2022; SIGGRAPH'22. [CrossRef]
16. Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Muller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. *arXiv* **2024**, arXiv:2403.03206.
17. Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.Y.; Ermon, S. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. *arXiv* **2022**, arXiv:2108.01073.
18. 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, Paris, France, 2–3 October 2023; pp. 3836–3847.
19. Guo, Y.; Yang, C.; Rao, A.; Wang, Y.; Qiao, Y.; Lin, D.; Dai, B. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv* **2023**, arXiv:2307.04725.
20. Khachatryan, L.; Movsisyan, A.; Tadevosyan, V.; Henschel, R.; Wang, Z.; Navasardyan, S.; Shi, H. Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. *arXiv* **2023**, arXiv:2303.13439.

21. Liu, X.; Gong, C.; Liu, Q. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In Proceedings of the The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, 1–5 May 2023.
22. Ceylan, D.; Huang, C.H.P.; Mitra, N.J. Pix2video: Video editing using image diffusion. *arXiv* **2023**, arXiv:2303.12688.
23. Wu, J.Z.; Ge, Y.; Wang, X.; Lei, S.W.; Gu, Y.; Shi, Y.; Hsu, W.; Shan, Y.; Qie, X.; Shou, M.Z. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 7623–7633.
24. Wang, W.; Xie, K.; Liu, Z.; Chen, H.; Cao, Y.; Wang, X.; Shen, C. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv* **2023**, arXiv:2303.17599.
25. Ronneberger, O.; Fischer, P.; Brox, T. Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015 Conference Proceedings, Singapore, 18 September 2022.
26. Yang, S.; Zhou, Y.; Liu, Z.; Loy, C.C. Rerender A Video: Zero-Shot Text-Guided Video-to-Video Translation. *arXiv* **2023**, arXiv:2306.07954.
27. Rowles, C. CiaraStrawberry/TemporalKit: An All in One Solution for Adding Temporal Stability to a Stable Diffusion Render via an Automatic1111 Extension. 2023. Available online: https://github.com/CiaraStrawberry/TemporalKit (accessed on 12 April 2023).
28. Face, H. Clip Interrogator: Prompt Engineering Tool That Combines OpenAI's CLIP and Salesforce's BLIP to Optimize Text Prompts to Match a Given Image. 2024. Available online: https://github.com/pharmapsychotic/clip-interrogator (accessed on 2 August 2024).
29. 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, 17–24 June 2023; pp. 22500–22510.
30. Li, J.; Li, D.; Xiong, C.; Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 12888–12900.
31. von Platen, P.; Patil, S.; Lozhkov, A.; Cuenca, P.; Lambert, N.; Rasul, K.; Davaadorj, M.; Nair, D.; Paul, S.; Berman, W.; et al. Diffusers: State-of-the-Art Diffusion Models. 2022. Available online: https://github.com/huggingface/diffusers (accessed on 2 August 2024).
32. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: New York, NY, USA, 2019; pp. 8024–8035.
33. Borji, A. Pros and cons of GAN evaluation measures: New developments. *Comput. Vis. Image Underst.* **2022**, *215*, 103329. [CrossRef]
34. Yang, S.; Jiang, L.; Liu, Z.; Loy, C.C. Pastiche master: Exemplar-based high-resolution portrait style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7693–7702.
35. Zhang, Y.; Huang, N.; Tang, F.; Huang, H.; Ma, C.; Dong, W.; Xu, C. Inversion-based creativity transfer with diffusion models. *arXiv* **2022**, arXiv:2211.13203.
36. Huang, N.; Zhang, Y.; Dong, W. Style-A-Video: Agile Diffusion for Arbitrary Text-based Video Style Transfer. *arXiv* **2023**, arXiv:2305.05464.
37. Qi, C.; Cun, X.; Zhang, Y.; Lei, C.; Wang, X.; Shan, Y.; Chen, Q. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. *arXiv* **2023**, arXiv:2303.09535.