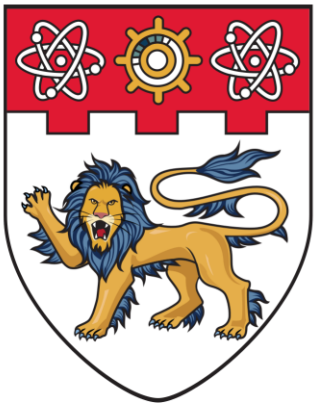


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Final Year Project

SCSE23-0532

Generative AI Art - Portrait Photography

Submitted By:

Ma Jiaxin
U2022011J

Supervisor: A/P Chia Liang Tien, Clement

Examiner: A/P Kwoh Chee Keong

Academic Year/ Semester: 2023/1

Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Computer Science of the Nanyang Technological University

Abstract

In recent years, the significance of artificial intelligence in art and image generation has become increasingly apparent, with the rise in popularity of AI image generators like DALL·E, MidJourney, and Stable Diffusion highlighting the myriad advantages and applications of automated image creation.

This project aims to explore the utilization of AI image generation techniques and develop a prototype image generator. Drawing upon insights gained from various computer science courses, the project will focus on designing a user-friendly web interface tailored specifically for photographers. The objective is to create an image generator capable of producing new images based on predefined parameters such as prompts, controNets, and styles.

In this project, the Stable Diffusion pipeline is utilized to design the customed web UI. This choice is made to leverage the robust capabilities of Stable Diffusion in image generation and to provide a seamless and efficient user experience for photographers. By integrating Stable Diffusion into the project, we aim to enhance the quality and versatility of the generated images while ensuring ease of use and accessibility for users.

Acknowledgements

I extend my sincere gratitude to several individuals whose support has been integral to the success of this project. Without their assistance, this endeavour would not have been possible.

Firstly, I am deeply thankful to Professor Chia Liang Tien, Clement, for his invaluable guidance and advice throughout the project. His patience and expertise were particularly crucial during the initial stages when I was uncertain about the project requirements. Regular meetings with him ensured we stayed on track, and I am grateful for the opportunity he provided me to undertake this project. Additionally, his assistance in arranging for us to borrow a functional workstation from the school for the project was greatly appreciated.

I am also appreciative of Chee Mei Qi, who, under the same professor, is working on a related project in AI Generative Arts but in a different area. Our collaboration has been seamless, with shared GPU resources from the school facilitating our work. I am thankful for her insights and advice during our meetings, and I am glad that our collaboration has been cooperative and constructive.

Lastly, I would like to express my gratitude to Takesawa Saori, also under the supervision of the same professor, for her valuable advice and guidance.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

In recent years, Generative AI Art has experienced a significant surge in popularity, leading to the emergence of a plethora of AI image generators in the market. Each of these generators showcases distinct strengths and capabilities. For instance, DALL·E 3 has gained recognition for its intuitive user interface, facilitating ease of use for a wide range of users. Midjourney, on the other hand, is renowned for its ability to produce high-quality image results, captivating artists and enthusiasts alike. Meanwhile, Stable Diffusion has carved out its niche by offering unparalleled customization and control over the generated images [1].

Despite its reputation for being less user-friendly compared to its counterparts, Stable Diffusion has garnered attention for its advanced features, notably its unique ControlNet functionality. This feature empowers users with granular control over the generation process, allowing for precise manipulation and customization of the output images.

1.2 Objectives

The primary objective of this project is to delve deeply into the capabilities of Stable Diffusion and explore its potential for generating portrait images. By conducting a thorough examination of its functionality and utilization, the project seeks to uncover the intricacies and challenges associated with using Stable Diffusion as a tool for image generation.

Furthermore, the project aims to address the usability concerns surrounding Stable Diffusion by developing a custom Web User Interface (UI) tailored specifically for photographers. This custom UI will be designed to simplify and streamline the complex process of utilizing Stable Diffusion, thereby enhancing its accessibility and user-friendliness. Through the design and unveiling of this custom UI, the project endeavors to make Stable Diffusion more approachable and accommodating for photographers and enthusiasts seeking to harness its advanced capabilities.

1.3 Scope

The scope of this project encompasses several key areas related to the exploration and utilization of Stable Diffusion for generating portrait images. Specifically, the project will:

- Conduct an in-depth analysis of Stable Diffusion's functionality and capabilities, with a focus on its unique ControlNet feature.

- Explore the intricacies and challenges inherent in using Stable Diffusion as a tool for image generation, particularly in the context of producing portrait images.

- Develop a comprehensive understanding of the various parameters and settings involved in the Stable Diffusion interface, and their impact on the generated images.

- Design and implement a custom Web User Interface (UI) tailored specifically for photographers, aimed at simplifying and streamlining the process of utilizing Stable Diffusion for image generation.

Chapter 2: Literature Review

This chapter focuses on explaining how stable diffusion operates. It will also give a thorough overview of the Diffusion model, explaining its fundamental concepts. Additionally, it will discuss the necessary computing resources required to effectively implement stable diffusion.

2.1 What is Stable Diffusion

Stable Diffusion stands out as a latent diffusion model designed specifically for generating AI images from textual prompts. Essentially, it operates as a text-to-image model: users input a text prompt, and in response, the model generates an AI-generated image that aligns with the provided text.

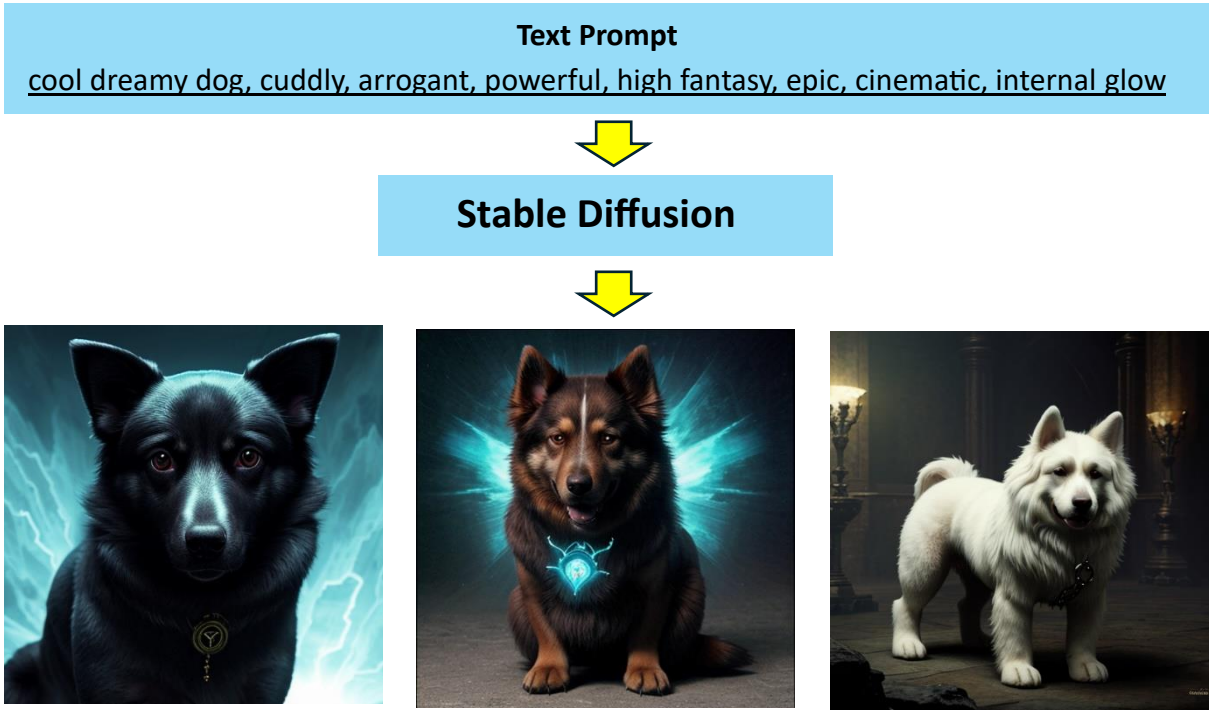


Figure 1: Stable diffusion turns text prompts into images.

2.2 Stable Diffusion model

Stable Diffusion is indeed an innovative approach within the realm of generative models, particularly within the domain of image generation. By compressing images into a lower-dimensional **latent space**, it manages to significantly reduce computational complexity while retaining the ability to generate high-quality images.

The key components of Stable Diffusion involve **forward diffusion** and reverse diffusion techniques. During training, forward diffusion is used to generate samples from a noise distribution, gradually adding noise to the input to create realistic images. **Reverse diffusion**, on the other hand, involves the process of generating high-quality images from their corresponding noise representations in the latent space.

By employing these techniques, Stable Diffusion effectively learns the underlying distribution of the training data in the latent space and can then generate novel images by sampling from this learned distribution. This approach offers advantages in terms of both efficiency and effectiveness compared to traditional high-dimensional image-based generative models. [2].

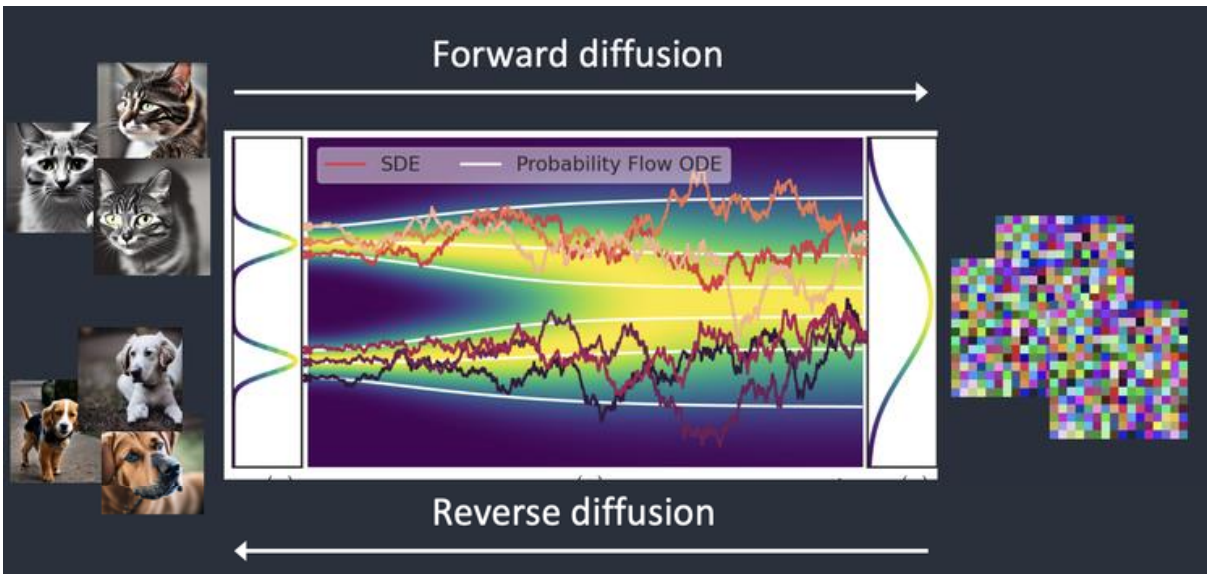


Figure 2: Forward and reverse diffusion of images [2]

2.2.1 Forward Diffusion:

Forward diffusion involves gradually adding noise to the training data. This incremental addition of noise aids the model in understanding the underlying distribution of the data by observing its changes as noise is introduced. Ultimately, this process leads to all the images resembling the final noised image.

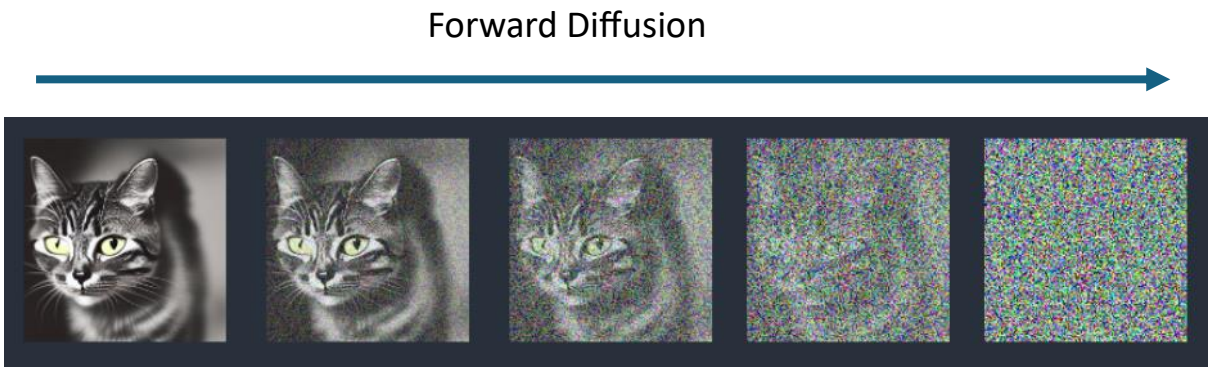


Figure 3: Forward and reverse diffusion of a cat [3]

2.2.2 Reverse Diffusion:

Reverse diffusion is a pivotal stage in the process where we generate a completely random image and then employ a noise predictor to identify the noise within it. This identified noise is subtracted from the original image. This process is repeated iteratively. Reverse diffusion plays a critical role in reconstructing the original data by reversing the noise added during the forward diffusion process. Essentially, it acts as a denoising mechanism to restore the original data from its noisy versions.

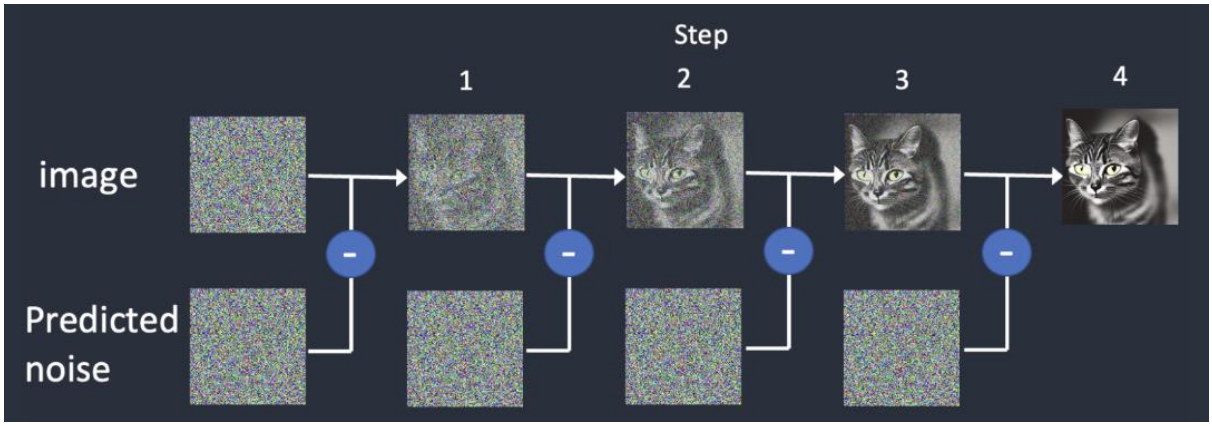


Figure 4: Reverse diffusion operates through iteratively subtracting the estimated noise from the image [3]

2.2.3 Latent Space

The Stable Diffusion model functions as a latent space model, employing a Variational Autoencoder (VAE) neural network consisting of an encoder and a decoder. The encoder condenses an image into a lower-dimensional representation within the latent space, while the decoder reconstructs the image from this latent space. Specifically, for a 512×512 image, the resulting latent image size is 4x64x64, which is 48 times smaller than the original image pixel space [4].

During training, rather than generating a noisy image directly, the model generates a random tensor in the latent space, known as latent noise. Instead of introducing noise directly to the image, it perturbs the representation of the image in the latent space with this latent noise. This approach is chosen for its efficiency, as operations within the smaller latent space are faster compared to those in the high-dimensional image space.

The forward and reverse diffusions are done in the latent space.

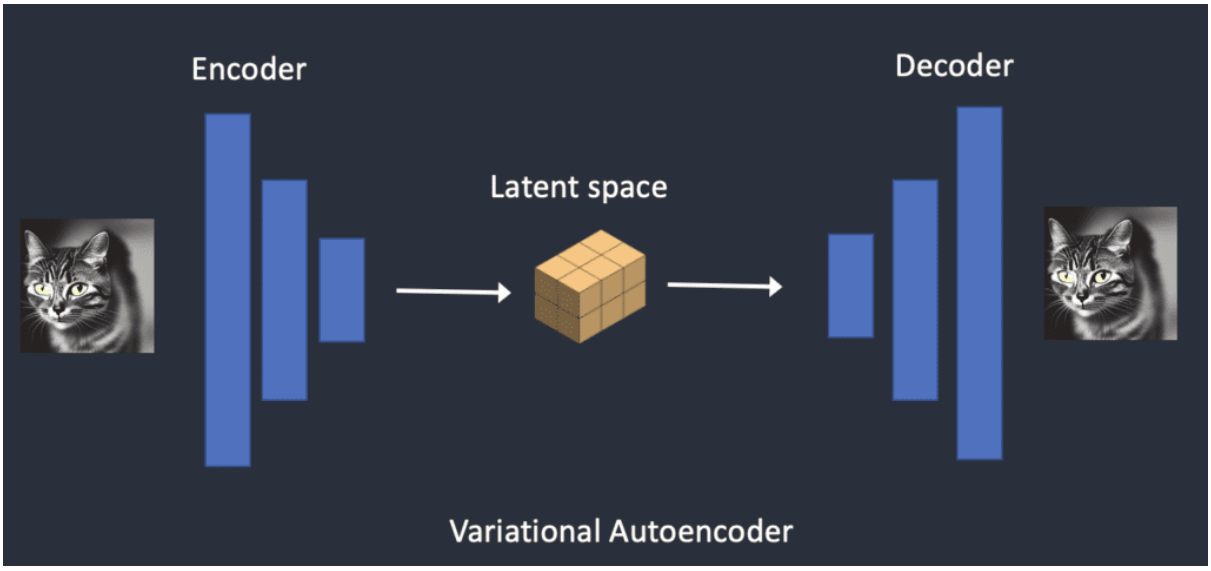


Figure 5: Variational autoencoder transforms the image to and from the latent space [3]

2.2.4 Training Procedure:

The training procedure consists of the following steps:

1. **Select Training Images:** Selecting a training image from a dataset. This image could depict a person, animal, or any other recognizable object. The choice of training images influences the diversity and quality of the generated data.

2. **Generate Random Noise Image:** Create a random noise image to introduce variability into the training process. This noise image serves as the basis for corrupting the original training image gradually.

3. **Corrupt Training Image:** Gradually corrupt the selected training image by adding the noisy image generated in the previous step (forward diffusion). This corruption process is typically performed over several steps, with each step introducing a controlled amount of noise to the original image. The gradual corruption helps the model understand and learn the underlying distribution of the data as noise is incrementally added.

4. **Train Noise Predictor:** Utilize the corrupted images to train the noise predictor model. The primary objective of the noise predictor is to estimate the noise level added during the corruption process accurately. This training involves adjusting the weights of the model through optimization techniques, such as gradient descent, to minimize the difference between the predicted noise level and the ground truth noise level provided during training. By iteratively training the noise predictor on a diverse set of corrupted images, the model learns to effectively estimate the amount of noise added during the diffusion process.

By following these steps, Stable Diffusion can effectively learn the relationships between noisy and original data, enabling the generation of

high-quality images during the reverse diffusion process. Additionally, training on a diverse dataset with various objects and scenarios helps ensure the model's ability to generate realistic and diverse outputs. Chapter 3 will present some of the training outcomes using portrait images.

2.3 Advantage and Limitation of Stable Diffusion

Advantages:

- **Enhanced stability and quality of image** results compared to traditional techniques like deep neural networks and Bayesian image analysis.

- Ability to generate **high-quality images from low-quality sources**, beneficial for enhancing old or low-resolution photos.

- Capability to enhance specific features in an image such as colours, textures, etc., **providing more control over the images**.

- Accessibility: Stable diffusion is freely available for everyone, although users need to run it on their own computers.

Limitation:

- **Computational Complexity:** The procedure comprises repetitive and sequential steps, rendering it computationally demanding. Reversing the noise introduced during forward diffusion demands substantial computational resources, particularly when handling large images [5].

- **Memory Consumption:** Operating at the pixel level necessitates storing and manipulating detailed pixel-level information during both forward and reverse diffusion. This can lead to high memory consumption, especially with high-resolution images, requiring efficient memory management strategies [5].

2.4 System Requirements

Unlike DALL·E, MidJourney, and many other generative AI art platforms that require a monthly subscription for full functionality due to the costs associated with generating images, Stable Diffusion provides all its features at no charge, as long as your computer meets the system requirements. However, it is important to note that limitations in computational complexity and memory consumption, as mentioned in section 2.3, can impact performance. Stable Diffusion relies solely on your computer's resources for image generation. Performance may vary based on the resources available. Here are the specifications for low-end, mid-range, and high-end systems [6].

LOW-END Specification

Component	Minimum Requirement
CPU	Quad-core processor
RAM	8GB
VRAM	4GB minimum, 6-8GB for better performance
Storage	50GB HDD or SSD

Table 1: Low-END Specification

Anticipated Performance and Constraints:

- Generation times are likely to be slower, particularly with higher resolutions.
- Lower resolution or simplified image complexity may be necessary for optimal results.

- Best suited for basic tasks and exploratory purposes.

MID-RANGE Specification

Component	Recommended Specifications
CPU	6-core processor
RAM	16GB
VRAM	10GB minimum, 10-16GB for improved performance
Storage	256GB SSD

Table 2: MID-RANGE Specification

Enhancements in Performance Compared to Low-End Specifications:

- Significantly quicker image generation coupled with enhanced resolution capabilities.
- Increased capacity to manage more intricate tasks with superior efficiency.

HIGH-END Specification

Component	High-Performance Specifications
CPU	8-core processor or higher
RAM	32GB or more
VRAM	16GB minimum, 16-24GB for the best performance
Storage	1TB NVMe SSD

Table 3: HIGH-END specification

Advantages of High-End Specifications:

- Remarkably accelerated generation times, ideal for high-resolution and intricate image generation.
- Augmented capability to execute multiple instances or manage demanding tasks concurrently.

Note: The mentioned specifications serve as guidance for users. As time progresses, stable diffusion technology evolves. New extensions may demand increased computational resources, or stable diffusion may improve to require even fewer resources.

2.5 Available Equipment Resources

The equipment resources utilized for this project are categorized as MID-RANGE specifications due to the VRAM capacity of 11 GB.

Component	Specifications
CPU	Intel(R) i7-8700K 6-core processor
RAM	32GB
VRAM	11GB
Storage	3TB SSD

Table 4: Specification used for the project

Chapter 3: Explorations

In this chapter, we explore the optimization of high-quality portrait image generation, which includes refining stable diffusion parameters, incorporating helpful extensions such as controlNet and roop, and integrating generative images with customized facial attributes. Practical experiments will be presented to enhance understanding of each concept.

All investigations are conducted using the Stable Diffusion Web User Interface (UI) named AUTOMATIC1111[7]. Installation guides are available in the repository located at: https://github.com/AUTOMATIC1111/stable-diffusion-webui

3.1 Stable Diffusion Parameters

Understanding and utilizing stable diffusion parameters is crucial for generating high-quality outputs. This section will begin by providing an overview of each parameter's function, followed by guidance on their proper utilization. Practical examples will then be presented to reinforce the learning process.

Please note that the parameters of Stable Diffusion are interconnected and work together. In the following explanations, the subsequent sections will endeavor to elucidate the usage of each parameter separately, supplemented with examples from experiments. It's important to understand how these parameters interact to influence the image

generation process. By examining each parameter individually, we can gain a deeper understanding of its role and significance. Moreover, accompanying examples from experiments will provide practical insights into how these parameters are applied in real-world scenarios. Through this exploration, we aim to provide comprehensive insights into the complexities of Stable Diffusion and its application in image generation.

3.1.1 Models

Stable diffusion encompasses numerous models trained by different individuals, available on reputable platforms like Hugging Face or CivitAI. It is crucial to rely on these trusted platforms, as other sources may pose security risks such as viruses. Each model is trained on distinct image datasets, making them proficient in different styles. For example, some excel at generating anime characters, landscapes, portraits, or paintings. Therefore, careful selection of a model is essential for generating desired images. Notably, if a model has not been trained with examples of cat images, it would not be capable of producing cat-like images.

Experiment 1: Examining the varying effects of different models on image generation

The images below originate from identical prompts but utilize various models. The "animaPencil" model embodies an anime aesthetic, "epicrealism" epitomizes realism, and "dreamshaper" captures the essence of realistic painting. Employing these models provides a straightforward method for achieving specific artistic styles.

Prompt: a man, smiling, full body shot, highly detailed, city street, wearing jeans

animaPencil

epicrealism

DreamShaper

Figure 6: Image generated using different model

Stable Diffusion v1.5 (SD1.5) VS Stable Diffusion Extra Large (SDXL)

All online models typically start with base models such as SD1.5 and SDXL. The key distinction lies in their training image sizes: SD1.5 is trained with 512x512 images, while SDXL is trained with 1024x1024 images. Due to its smaller image size, SD1.5 requires significantly lower computational resources compared to SDXL. Consequently, SD1.5 models are recommended for low-end and mid-range specifications, while SDXL is suitable for mid-range to high-end specifications. Despite requiring higher computational resources, SDXL can generate higher resolution images.

The optimal resolutions for these models are as follows:
SD1.5: 512x512, 768x768, 512x768, 568x512, 640x832, 832x640
SDXL: 1024x1024, 896x1152, 1152x896, 768x1344, 1344x768, 640x1536, 1536x640

If the generated image does not adhere to the optimal resolution size, duplication issues may arise because the window sizes for SD1.5 and SDXL are fixed. In such cases, larger images should be upscaled using an upscaler after the image is generated.

However, since SD1.5 requires fewer computational resources, it hosts a greater number of trained models online compared to SDXL. The choice between SD1.5 and SDXL models ultimately depends on the specific requirements and computational resources available.

3.1.2 Prompt

Crafting a robust and effective prompt is essential for producing top-notch images. Here are some strategies to enhance the quality of generated images:

I. Subjects: Ensure clarity and specificity in describing the subjects of the image.

II. Negative prompt: Specify exclusions to guide the model away from undesired outcomes.

III. Keyword weights: Adjust the importance of keywords to influence the image generation process.

IV. Style: Define the artistic style desired for the image, including options such as painting, surrealism, and pop art.

Each strategy plays a crucial role in refining the image generation process and achieving desired results.

I. Subjects

Ensure an adequate variety of subjects and punctuate each one with a comma for clarity (The model utilized for this demonstration is epicrealism).

Experiment 2: Exploring the effective utilization of subjects in generating images.

From the image generated below, it is apparent that incorporating subjects can enhance the overall composition significantly (The model used for this demonstration is epicrealism).

Prompt: a man

Prompt: a man, smiling, full body shot, highly detailed, city street, wearing jeans

Figure 7: how to utilize subjects

II. Negative Prompt

useful for specifying what not to include in an output or for directing the model away from certain undesired outcomes. They can be employed to refine the generation process, particularly when seeking specific results.

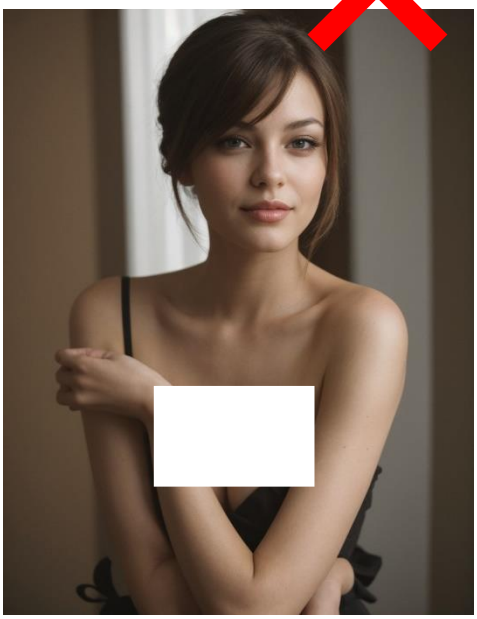
Stable diffusion models may occasionally generate nude and sexy photos (especially for woman images) or extra limbs, but negative prompts significantly assist in mitigating these undesirable effects.

Experiment 3: Investigating the effective implementation of negative prompts in image generation.

The images below exemplify the application of negative prompts, specifically targeting content not safe for work (NSFW), and demonstrate their effect on the generated images. The generator has filtered out elements that match the negative prompt (The model used for this demonstration is epicrealism).

Prompt: A beautiful woman

Negative Prompt:



Prompt: A beautiful woman

Negative Prompt: nsfw

Figure 8: how to utilize negative prompt nsfw

Experiment 4: Utilizing negative prompts to eliminate mutated hands.

The following image demonstrates how a negative prompt assists in enhancing the image by transforming a mutated hand to appear normal (The model utilized for this demonstration is epicrealism).

Figure 9: Improvements on negative prompt

Universal Negative Prompt:

Various images generated may exhibit different issues, but employing a negative prompt allows us to eliminate these problems. However, since we do not know the specific issues beforehand, a universal negative prompt can be employed to address this uncertainty. The universal negative prompt is outlined below.

Negative Prompt: nsfw, deformed, mutated, mutilated, distorted, disfigured extra limbs, missing limbs, floating limbs, disconnected body parts, missing

arms, missing legs, missing fingers, amputated, amputation, extra arms, third arm, extra legs, too many fingers, fused fingers, low quality, low resolution, pixelated, jpeg artifacts, blurry, unclear, out of focus, depth of field, bad anatomy, wrong anatomy bad proportions, disproportionate

Despite the effectiveness of the universal negative prompt mentioned above, it should be customized to fit one's preferred image generation. For instance, users may wish to add exclusions for anime or cartoons if they do not want to see them or remove options like "blurry" or "out of focus" if they desire to generate images with a Bokeh effect.

III. Keyword Weight

The weight of a keyword can be adjusted using syntax in two ways. First, through the format (keyword: factor), where the factor, a numerical value, signifies importance; values less than 1 denote decreased significance, while those greater than 1 indicate increased importance. Second, by adding multiple + or - symbols behind the keyword, with + augmenting its importance and - diminishing it. Third, using multiple Parentheses () can also achieve similar adjustments in keyword weighting.

Weight conversion between (keyword: factor) and (Keyword) symbols[8]:

- + is equivalent to 1.1, ++ to 1.1^2, +++ to 1.1^3, etc.
- - is equivalent to 0.9, -- to 0.9^2, --- to 0.9^3, etc.
- () is equivalent to 1.1, (()) to 1.1^2, etc.

Experiment 5: Employing weight control to customize the image.

The following outputs showcase different images achieved by adjusting their respective weights (The model utilized for this demonstration is epicrealism):

Prompt: a bear eating apple pie

Prompt: a (bear:1.1) eating apple pie

Prompt: a (bear:1.3) eating apple pie

a bear (eating)+++ apple pie

Prompt: a (bear eating)+++++ apple pie

Prompt: a bear eating (((apple))) pie

Figure 10: Adjusting the weight of the prompt

Analysis:

Image 1 does not portray the intended scenario of "a bear eating apple pie." Instead, it features a girl wearing bear ears while seemingly posing for a photo with an apple pie.

In Image 2, we increased the weight of the keyword "bear" to 1.1. Here, the bear is incorporated into the design of the apple pie. The improvement from Image 1 to Image 2 lies in the absence of the girl and the presence of a bear, albeit as part of the pie's design.

In Image 3, we further increased the weight of "bear" to 1.3, resulting in a clearer depiction of a bear. However, the bear is not shown eating the apple pie.

In Image 4, we adjusted the weight of the keyword "eating," leading to the depiction of a girl eating the apple pie.

In Image 5, we significantly increased the weight of "(bear eating)+++++," equivalent to "bear eating : 1.1^5." As a result, the desired image of a bear eating apple pie appears.

In Image 6, we addressed the fact that apples are typically not visible in apple pie due to being smashed into filler. By increasing the weight of the keyword "apple," the generated image displays apples.

IV. Style

The term "style" encompasses the artistic expression depicted in the image, spanning various genres such as painting, surrealism, pop art, and more. Moreover, one can introduce the name of a movie or article to alter the image style, especially if the model has been trained to replicate those specific styles.

Experiment 6: Exploring the use of styling in image generation.

The following images illustrate how different artistic styles influence the appearance of the images (The model utilized for this demonstration is realvisxlV40).

Prompt: a man holding a sword, cyberpunk

Prompt: a man holding a sword, realistic

Prompt: a man holding a sword, water colour

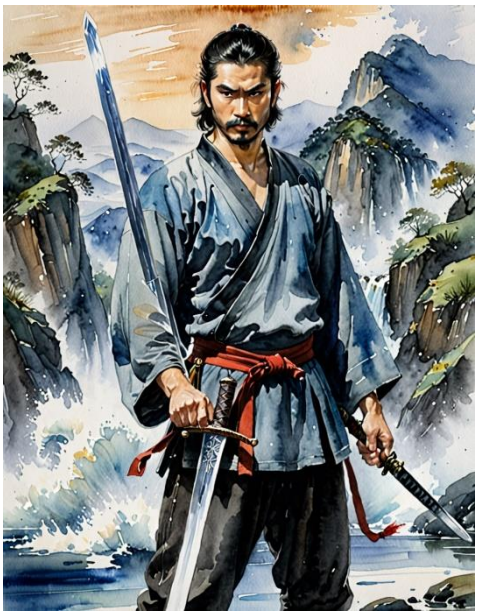


Figure 11: Image Styling effect

Experiment 7： Investigating scenarios where styling does not produce the desired effects.

The image exemplifies that when a model is specialized in producing exclusively realistic images, attempting to apply different styles yields consistent output resembling the model's inherent style. As elucidated in session 3.1.1 on models, this particular model is not trained to accommodate other styles. (The model utilized for this demonstration is epicphotogasm).

Prompt: a man holding a sword, cyberpunk

Prompt: a man holding a sword, realistic

Prompt: a man holding a sword, water colour

Figure 12: Scenarios where image styling fails to produce the desired effect

From Experiment 6 and 7, it is evident that choosing a suitable model is crucial. Different models are trained on diverse images and styles, so selecting the appropriate one is essential. Additionally, mastering how to

incorporate the desired style effectively within the prompt is vital for achieving the desired effect.

3.1.3 Sampling Methods and Steps

In session 2.2, it is established that Stable Diffusion initiates the image production process by generating a completely noise image within the latent space. Subsequently, the noise predictor estimates the image's noise.

In session 2.2.2, Reverse Diffusion, this estimated noise is then subtracted from the image. The iterative procedure is repeated multiple times to refine the image and remove noise.

Furthermore, the denoising process, also known as sampling, involves generating a new sample image at each step by Stable Diffusion. The technique utilized during this sampling process is referred to as the sampling method [9].

In image generation, different sampling methods vary across various aspects:

I. Noise Source: These methods utilize different types of noise, such as Gaussian or uniform noise.

II. Noise Injection: They vary in how they introduce noise into images, whether directly into pixels or intermediate representations.

III. Sampling Strategy: Methods differ in their approach to sampling, including sequential, parallel, or batch sampling techniques.

IV. Iteration Scheme: The number of iterations and convergence criteria vary, influencing the speed and quality of convergence.

V. Noise Modelling: Methods employ diverse models to estimate and remove noise, ranging from simple techniques to more advanced deep learning models.

VI. Computational Efficiency: There are differences in computational requirements, with some methods being more resource-intensive but offering higher image quality.

Overall, the choice of sampling method significantly impacts image quality, diversity, and computational efficiency, making certain methods more suitable for particular image types.

Experiment 8： Comparing different sampling methods.

From the images below, it is evident that despite employing the same number of sampling steps, sampling methods like DPM++ 2M Karras, Euler, and Heun excel in effectively removing noise, resulting in clear images. Conversely, methods such as DPM fast, PLMS, and LMS still exhibit residual noise. These variations highlight how different sampling methods handle noise removal differently, ultimately leading to distinct visual outcomes in the images (The model utilized for this demonstration is epicrealism).

Prompt: a man holding a sword

Sampling Steps: 24

DPM++ 2M Karras

Euler a

Heun

DPM fast

PLMS

LMS

Figure 13: Comparing different sampling methods

Experiment 9: Investigating the impact of sampling steps on image generation.

The following images illustrate how changing steps impacts the generated images across various samplers. (The model utilized for this demonstration is epicrealism)

Figure 14: Modifying the sampling method and adjusting the steps can influence the resulting images

Analysis:

From the images depicted in Figure 14, it is evident how the steps impact the generated images, progressing from step 1 to step 40. As the steps increase, the image resolution improves, and noise decreases. Additionally, we can assess the performance of each sampler. Most samplers begin with sample images except PLMS, which starts with a completely noisy image at step 1. DPM++ 2M Karras and Heun produce similar images due to consistent lighting, while Euler-A generates darker images compared to other samplers.

Despite DPM Fast producing more complex backgrounds, LMS sampler performs the worst, with noticeable noise in the images even at step 40. Euler-A and Heun perform exceptionally well, generating high-quality images around 10 steps.

Overall, DPM++ 2M Karras, Euler-a, and Heun are recommended for most users, as higher sampling steps require more computational resources. It is important to note that different samplers offer varying lighting conditions, so choose accordingly based on your requirements.

3.1.4 Classifier Free Guidance (CFG) scale

The CFG scale is a parameter that governs Stable Diffusion, determining how closely it adheres to the input prompt during image generation. Lower CFG values grant the AI more freedom for creativity, potentially resulting in deviations from the prompt. Conversely, higher CFG values enforce stricter adherence to the prompt [10].

- CFG 2-6: Offers creativity but may lead to significant distortion and departure from the prompt. Suitable for brief prompts and experimentation.

- CFG 7-10: Recommended for most prompts, striking a balance between creativity and guided generation.

- CFG 10-15: Ideal when prompts are highly detailed and specific about the desired image outcome.

- CFG 16-20: Not recommended unless the prompt is exceptionally detailed, as it may compromise coherence and image quality.

Experiment 10: Exploring the applications of CFG scale in image generation.
The image below demonstrates the impact of the CFG scale on the resulting image and underscores the significance of selecting an appropriate CFG scale.

Prompt: A playful red panda skateboarding down a neon-lit street in Tokyo, wearing a tiny helmet and backpack, high detailed, dynamic composition, vibrant colors. (The model utilized for this demonstration is realvisxlV40)

Figure 15: The CFG Scale parameter directly impacts the image generated

From the images depicted in Figure 15, the influence of CFG Scale on the generated image is evident. At CFG Scale 2, the panda is depicted without wearing a tiny helmet. However, as the CFG Scale increases to 4, the panda appears with its helmet. With higher CFG values, the image tends to adhere more closely to the prompt, but this may come at the expense of its ability to generate diverse and high-quality images. Furthermore, as the CFG Scale increases, the colour gradient in the image becomes less smooth and more uniform.

3.1.5 Seed

A seed number forms the basis of the image generation process. By maintaining a consistent seed number across experiments, Stable Diffusion ensures that despite varying parameters such as the model, sampling method, steps, and CFG, the generated images remain identical. However, when generating a random image, a seed number of -1 is employed. This uniformity in seed usage ensures that images across experiments exhibit similarities, facilitating effective comparison and analysis.

Experiment 11: Investigating the utilization of the seed in image generation.

The images presented in experiments 3 and 4 were generated using a seed of -1, which explains their differing appearances.

Prompt: A beautiful woman

Negative Prompt: nsfw

Prompt: A beautiful woman

Negative Prompt: nsfw

Figure 16: different seed generate different image

The seed number can be found in the information banner located at the bottom of the generated image. It's worth noting that the seed number for both images is identical.

In Experiment 4, we employed the seed number to maintain the overall structure of the image while specifically focusing on correcting mutated hands to appear normal. Additionally, we utilized negative prompts to guide the image generation process. However, to avoid any confusion, Experiment 4 primarily centred on explaining the usage of negative prompts and did not delve into the specifics of seed utilization. It is important to note that in all other experiments, a consistent seed number was indeed employed to ensure similarity among the generated images. Furthermore, ControlNet

was also utilized to maintain the posture of the lady in the image, a technique that will be introduced in next session.

The seed number can be found in the information banner located at the bottom of the generated image. It's worth noting that the seed number for both images is identical. Additionally, in the amended image, ControlNet is utilized to maintain the same posture and clothing.

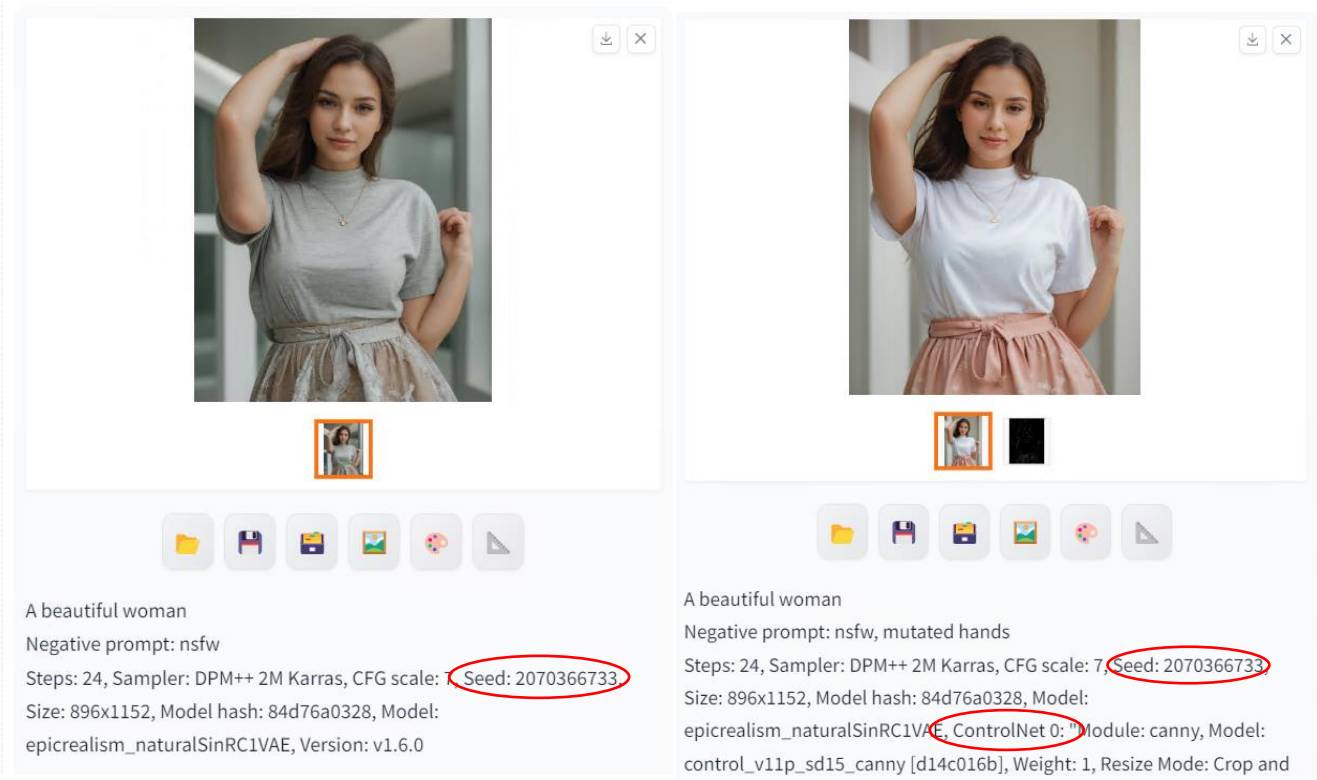


Figure 17: where to find the seed number

3.1.6 Low-Rank Adaptation (LoRA)

LoRA (Low-Rank Adaptation) is a training technique utilized for fine-tuning Stable Diffusion models. It can be applied in conjunction with SD1.5 and

SDXL models, allowing for the fine-tuning of images using multiple instances as desired.

LoRA is implemented in the prompt after downloading the LoRA models and saving them to the LoRA model directory.

Experiment 12: Generate image with LoRA

No LoRA

Prompt: an portrait of an female doctor drinking coffee, detailed, centered, digital painting, artstation, concept art, donato giancola, Joseph Christian Leyendecker, WLOP, Boris Vallejo, Breathtaking, 8k resolution, extremely detailed, artistic, hyperrealistic, beautiful face, octane render, fallen satellite, twisted metal, electronic glitches, lost connections

LoRA

Prompt: an portrait of an female doctor drinking coffee, detailed, centered, digital painting, artstation, concept art, donato giancola, Joseph Christian Leyendecker, WLOP, Boris Vallejo, Breathtaking, 8k resolution, extremely detailed, artistic, hyperrealistic, beautiful face, octane render, fallen satellite, twisted metal, electronic glitches, lost connections <lora:detailed_notrigger:1>

Figure 18: Compareing images use LoRA

The last part of the prompt indicates the use of LoRA. It can be placed anywhere in the prompt. Both images use the same prompt, with the first one lacking the LoRA statement, while the second one includes the LoRA statement "<lora:detailed_notrigger:1>". The "1" in the statement indicates the weight, with higher numbers indicating more emphasis on LoRA. As a result, the second image exhibits more details overall, particularly noticeable in the clarity of the printings on the cups and the hair, which appears much more detailed with LoRA. Additionally, the background features more lamps in the image where LoRA is used.

It is worth mentioning that different LoRA models are trained for specific purposes. For instance, some are tailored for drawing better eyes, while others specialize in figures or anime. The LoRA model used here, "detailed_notrigger," is particularly suited for adding intricate details, especially in anime-style drawings. Consequently, the image with LoRA looks more like a drawing compared to the original, which appears more realistic.

3.2 ControlNet

ControlNet, an extension of Stable Diffusion, serves as a neural network framework aimed at regulating diffusion models by imposing additional constraints. One notable capability of ControlNet is its ability to generate an image map derived from an existing image, providing precise control over composition and human poses in AI-generated images.

To effectively utilize ControlNet, it's essential to download the appropriate ControlNet models. While both SD1.5 and SDXL require different ControlNet models, the usage process remains consistent. However, SDXL offers a more limited selection of ControlNet models compared to SD1.5, with only three models available: Canny, OpenPose, and Depth. In contrast, SD1.5 provides access to over 10 ControlNet models.

Furthermore, ControlNet includes a Control Mode parameter, which regulates the balance between the significance of the prompt and the importance of ControlNet during image generation. The upcoming experiment will explore the variations that arise when different Control Modes are employed.

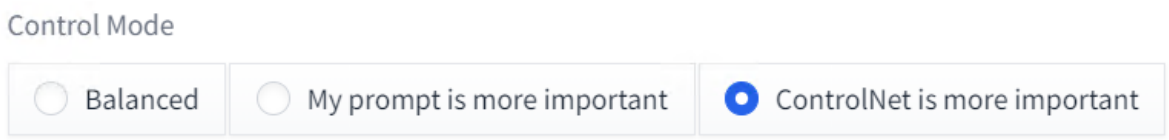


Figure 19: Control Mode

Experiment 12: Generating images using ControlNet.

The ControlNet extension initially generates a ControlNet image as the base image. The images below showcase ControlNet images generated by various ControlNet models (The model used for this demonstration is epicrealism).

For the sake of fairness in testing, all other settings will remain constant:

Pemopt: a man wearing a cap, sunflower field

Model: dreamShaper_8

Sampling method: DPM++2M

Sampling Steps: 40

CFG Scale: 7

The following image will function as the input image for generating all ControlNet images.

Figure 20: Input controlNet Image

ContorlNet: Canny

In the Canny scenario, the ControlNet image is generated by detecting the edges from the initial image.

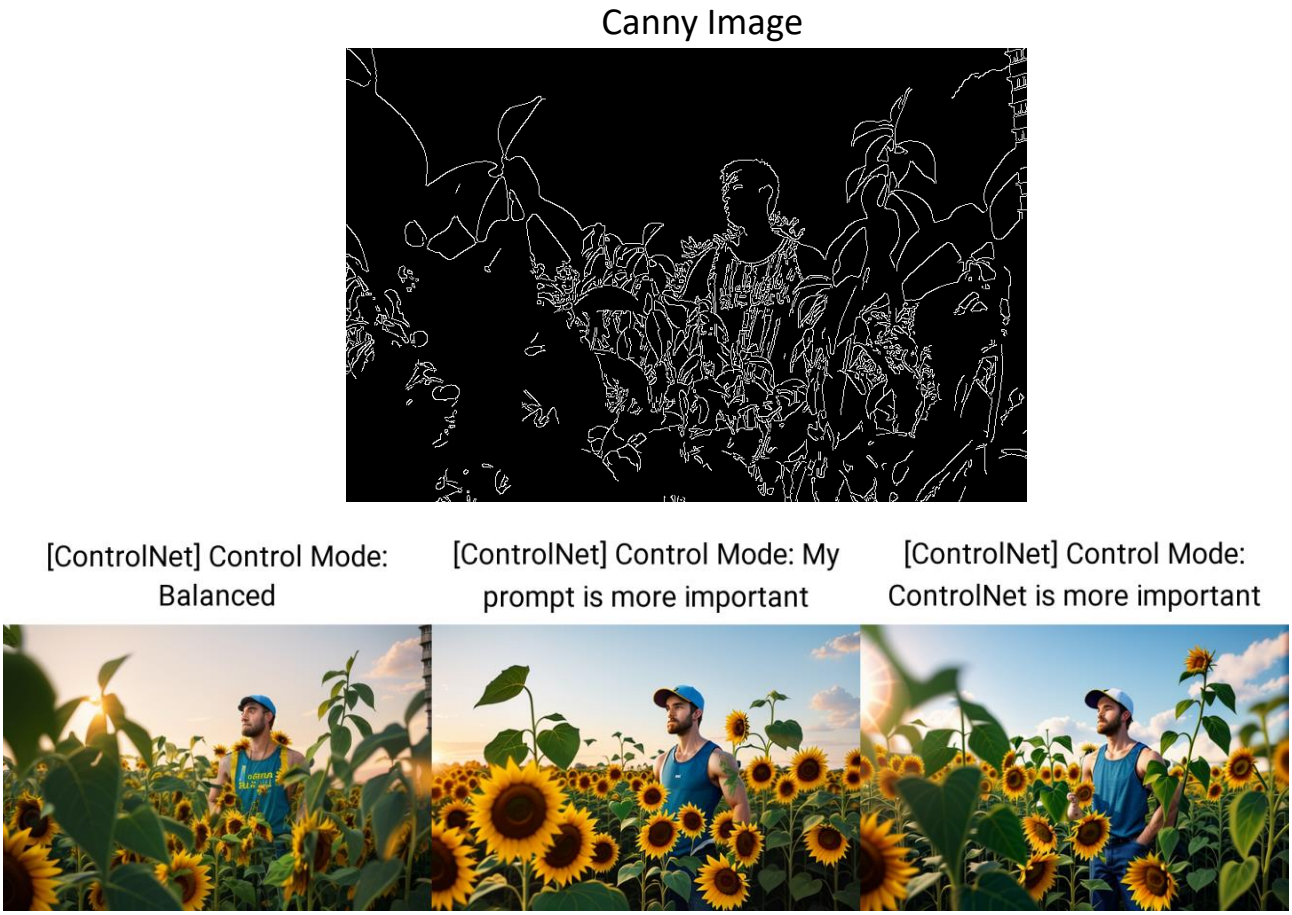


Figure 21: Imaged generated using ControlNet

From the output image above, it is evident that both the "My prompt more important" and "Balanced" modes yield more natural-looking results compared to the "ControlNet more important" mode. In the "ControlNet more important" mode, efforts to align with the ControlNet images have introduced some unusual details. Notably, all three images depict individuals wearing singlets, consistent with the ControlNet images provided.

ContorlNet: Depth

In the depth scenario, the ControlNet image is generated based on the depth information extracted from the original image.

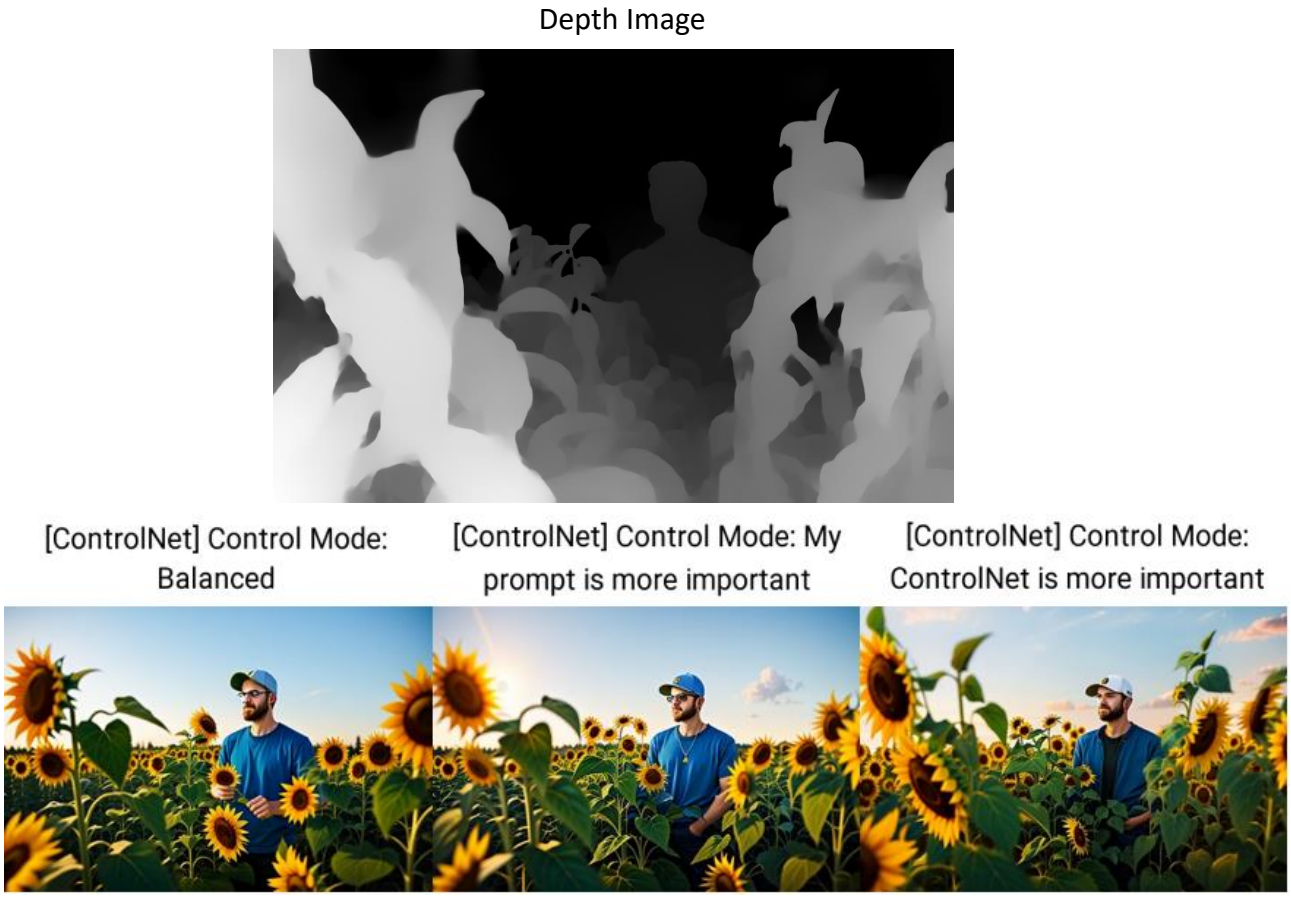


Figure 22: Imaged generated using Depth ControlNet

In the generated output image, the layout remains consistent across all ControlNet modes. However, in the "ControlNet more important" mode, attempts to align with the ControlNet images have resulted in the introduction of some unusual details. Notably, the attire of the person in the image no longer includes a singlet. This discrepancy arises because the Depth ControlNet does not capture such fine details.

controlNet: OpenPose

In the OpenPose scenario , the ControlNet is specifically designed to detect and analyse human body postures.

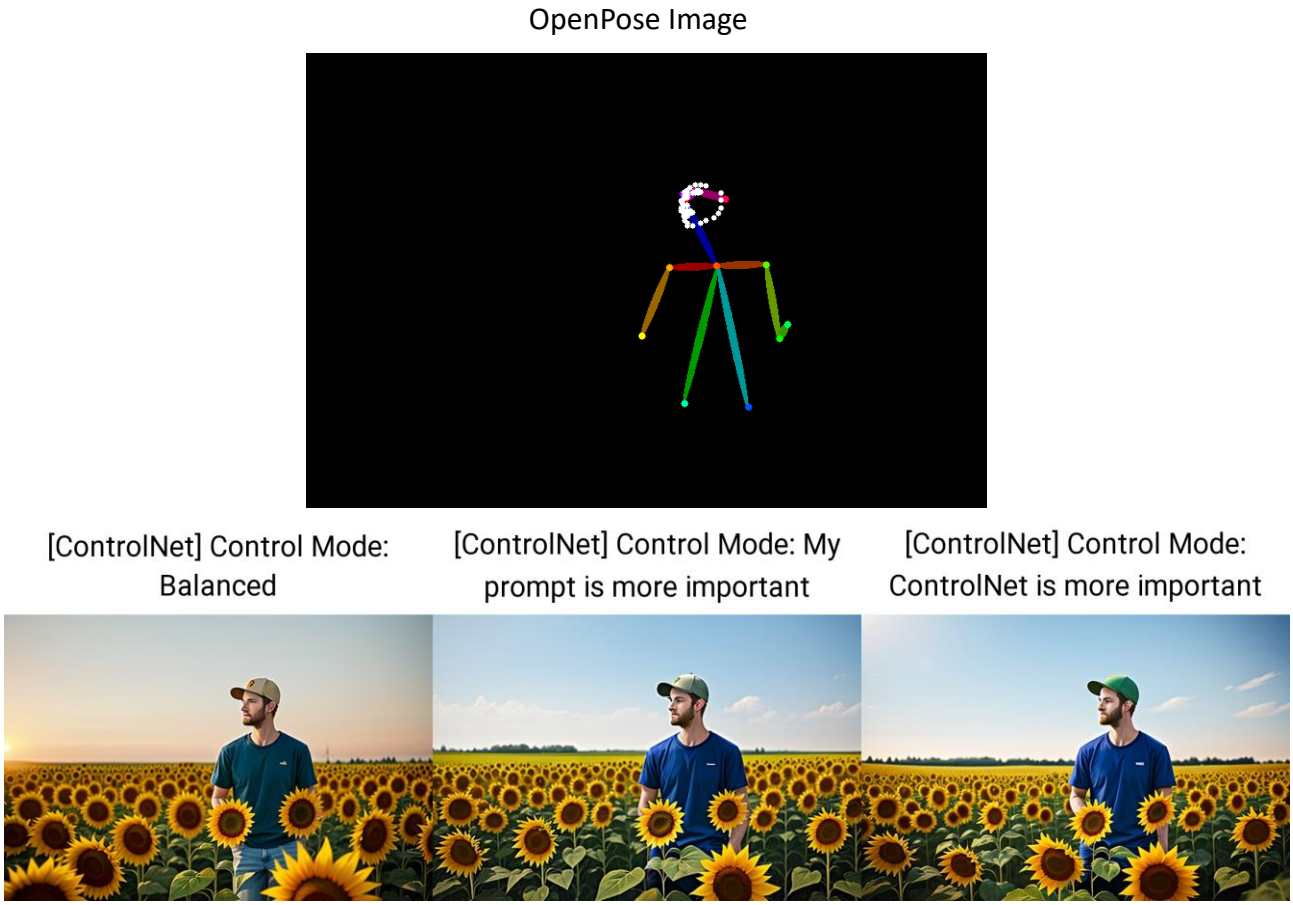


Figure 23: Imaged generated using OpenPose ControlNet

In the generated output images, it is evident that the person is not wearing a singlet, and the background differs significantly from the original image. This discrepancy arises because OpenPose only captures the human posture and does not consider other elements such as clothing or background details.

Limitation on OpenPose ControlNet:

Images without humans cannot effectively utilize OpenPose ControlNet. The following image demonstrates that OpenPose fails to capture the posture of the cat.

Image for generate controNet image

OpenPose Image

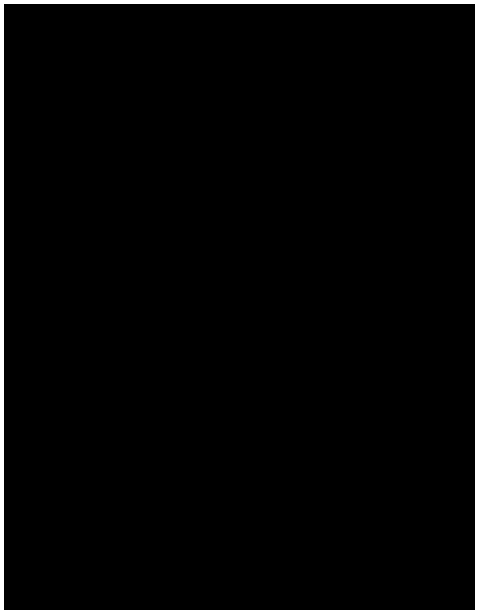


Figure 24: non-human image generates empty openPose image

ControlNet: Others

As previously stated, while SDXL models offer ControlNets for Canny, Depth, and OpenPose, SD1.5 provides a broader selection of ControlNet options. The following images will exclusively showcase the ControlNet images generated without providing detailed explanations for each one, serving the purpose of showcasing their variety.

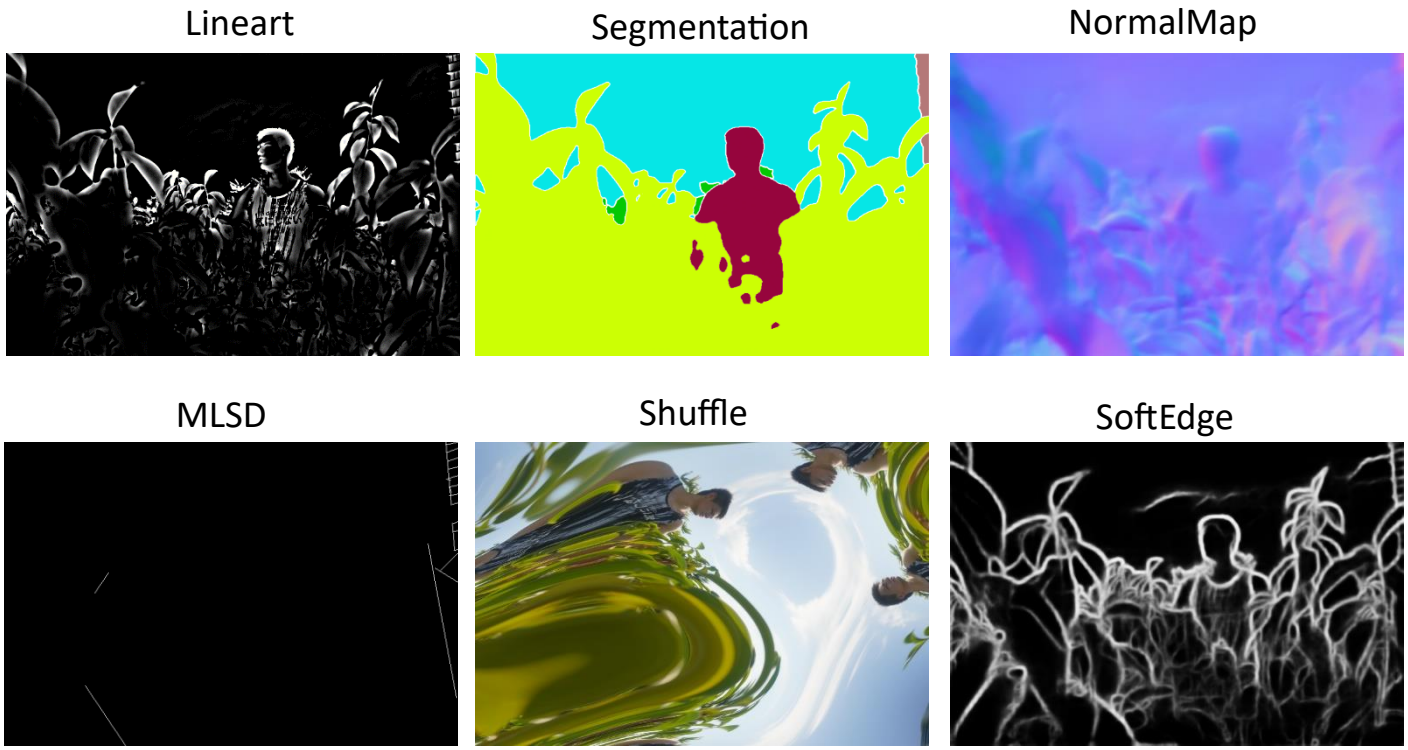


Figure 25: Other ControlNet Images

From all the ControlNets available, it's evident that each ControlNet excels in different aspects. Canny, Lineart, and Soft Edge ControlNets are proficient at outlining the image, providing clear delineation of objects and shapes. On the other hand, Depth, Segmentation, and NormalMap ControlNets excel at accurately placing the depth of objects within the image, enhancing spatial perception. MLSD ControlNet specializes in drawing straight lines with precision, while Shuffle ControlNet is adept at randomizing images, introducing variability and diversity in the generated outputs.

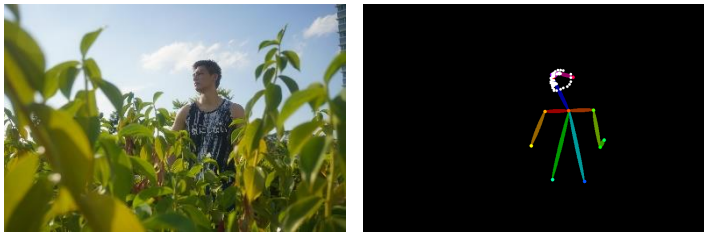
Experiment 13: Generating images using multiple ControlNet.

Multiple ControlNets can be employed to generate images, with a recommendation of using up to three ControlNets simultaneously. Using

more than three ControlNets may overly constrain the AI's ability to generate images, limiting its creativity and potential for diverse outputs.

The following example demonstrates the practical application of using multiple ControlNets. The settings used are consistent with those in Experiment 12.

ControlNet 1: OPenPose



ControlNet 2: Shuffle

ControlNet 3: Depth

Resulting image of ControlNet 1 & 2

Resulting image of ControlNet 1 & 2 & 3

Figure 26: Utilizing multiple ControlNet

In the first image, two men are depicted, with the second man positioned at the location specified by ControlNet 1 OpenPose image. The appearance of the first man is influenced by ControlNet 2 Shuffled image, resulting in the presence of two heads. Additionally, the plants in the image are not as tall as in the original images.

The second image showcases a combination of all three ControlNets. The presence of flying leaves is attributed to ControlNet 3 Depth of Flying Leaves image. The man is positioned based on the OpenPose image location, with ControlNet 2 further contributing to the randomness of the image. Again, the plants are not as tall as in the original images. When multiple ControlNets are utilized, the AI strives to strike a balance among them.

3.3 Face Replacement

There are three primary methods for generating your face using Stable Diffusion: retraining the model, inpainting your face into the image, and utilizing the Roop extension. Roop extension, a relatively new addition to Stable Diffusion, has shown superior performance compared to the other two methods. In this evaluation, we will assess all three methods, examining their respective advantages and disadvantages.

3.3.1 Training Images into the Model

Training Your Own Image into the Model with DreamBooth: A Step-by-Step Guide

I. Collect Images: Gather a minimum of 10 images featuring the subject or style you wish to train into the diffusion model.

II. Create Text Files: Prepare text files corresponding to each image, providing detailed descriptions of the content and characteristics of each image.

III. Select a Base Model: Choose a base model suitable for your training images and objectives.

IV. Utilize DreamBooth: Use DreamBooth, a training technique that updates the entire diffusion model by training on a small set of images, to train your selected images into the model.

The individual depicted in the following images is the person I trained the model for. All the images are described as "a photo of Jiaxin."

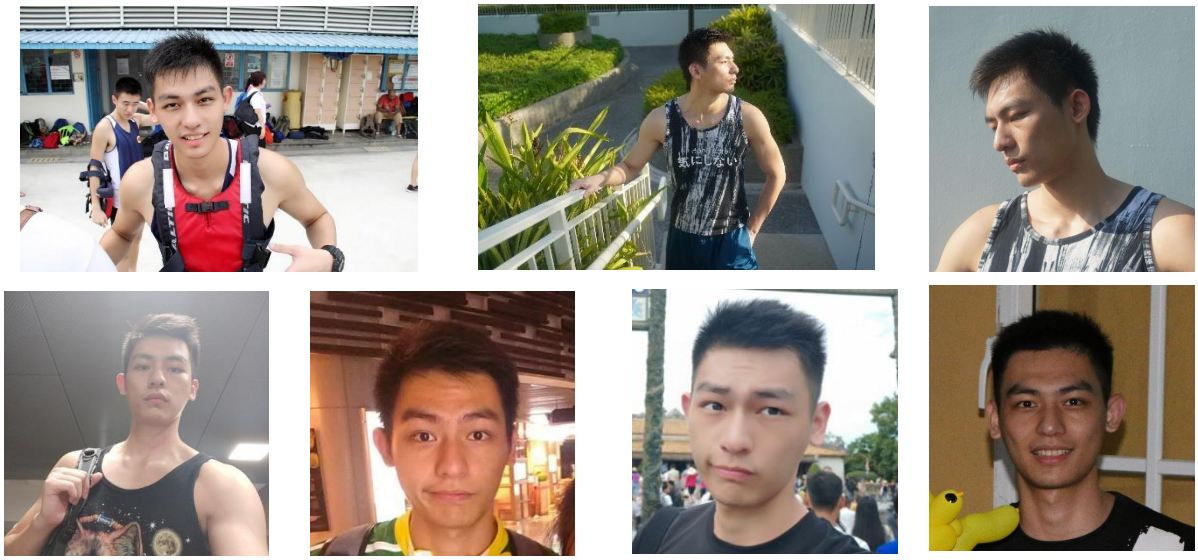


Figure 27: Image of Jiaxin

Experiment 14: training Images of Jiaxin

Training 1

No. of photo: 20 different images of Jiaxin

Result:

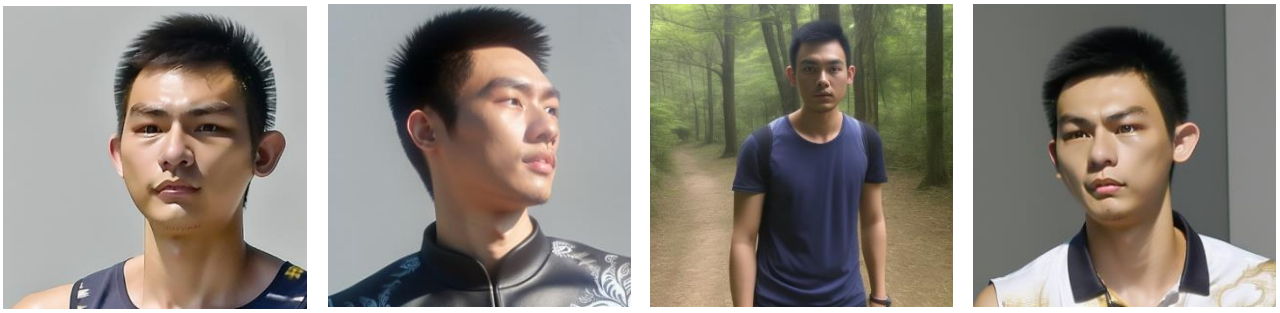


Figure 28: 1st training results

The generated image closely resembles the trained images of the person; however, there are numerous inconsistencies and anomalies present.

Training 2

No. of photo: 40 different images of Jiaxin

Result:

Figure 29: 2nd training result

The generated image has significantly improved.

As the number of training images increases, the trained model tends to improve. To achieve optimal results, thousands or even millions of photos are required. However, this necessitates significant computational resources and longer training times. Consequently, this method is not recommended for individuals with limited computational resources and is generally not suitable for widespread use by the public.

3.3.2 Inpainting

Inpainting is a technique used to replace or reconstruct an object within an image seamlessly. In the context of face swapping, inpainting is employed to replace the face of an individual within an image, particularly in the designated mask area where the original face is located. This process ensures that the replacement face blends naturally with the surrounding features and maintains the overall integrity of the image.

Expriment 15: Face Swapping via Inpainting

The image below serves as the primary image, and the individual within it will undergo masking for the purpose of conducting a face replacement experiment.

Figure 30: Inpainting primary image

Step 1: Utilize ControlNet to generate an image, ensuring the person's position in the primary image is accurately fixed. Failure to do so will result in the technique not functioning properly, as demonstrated in the failed examples to be shown later.

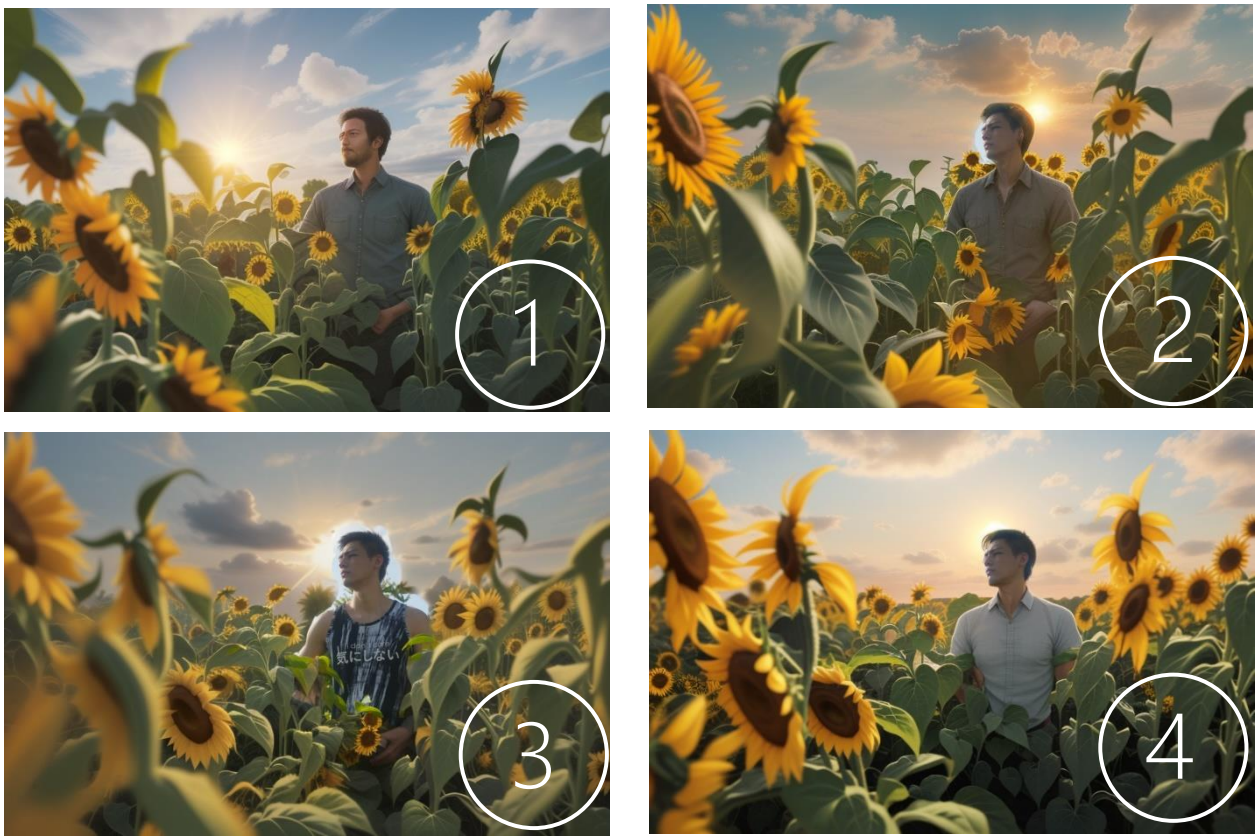


Image 1 displays the generated image using ControlNet, where we intend to perform a face swap for the man depicted.

Image 2 highlights a noticeable color difference near the face due to the excessively large masked area.

Image 3 exemplifies an instance of inpainting the entire person into the image, with the masked area clearly visible.

Image 4 showcases a near-perfect result, with careful masking of the area around the person's face. Upon zooming in, slight differences may be observed along the edges of the face.

Figure 31: Zoom in of image 4

Upon zooming in on Image 4, we can discern a drawback of using inpainting for face replacement. At higher magnification levels, it becomes evident that the image is composed of elements from two distinct images

The image below illustrates a failed attempt at inpainting without utilizing ControlNet. It is clear that the masked face has been overlaid onto the girl's face.

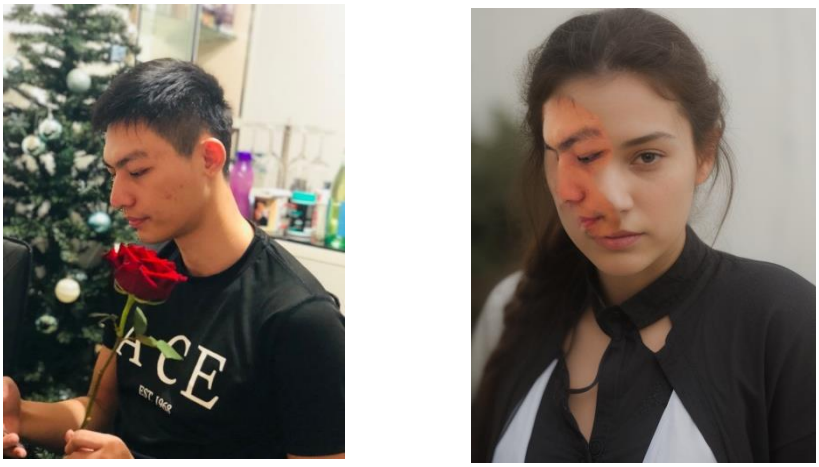


Figure 32: Inpainting without ControlNet

3.3.3 Roop

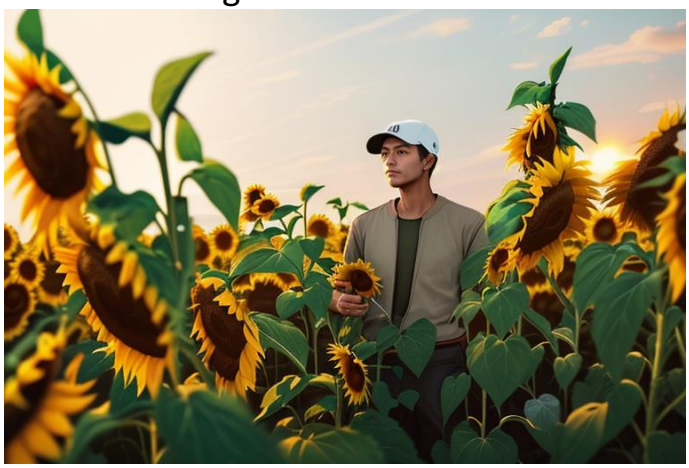
Roop extension is specifically designed for face swap, aiming to address the challenges associated with face replacement. It eliminates the need for time-consuming image training while ensuring smooth image generation.

Roop can be utilized both with and without ControlNets.

Expriment 16: Face Swapping via roop

Image used for roop

Image Generated



Zoom in Image

Figure 33: Imaged generated using roop

Form the above imaged we can see that the face swap is about 80% looks the same, and the entire image is looks smooth, it does not work well if the input face is too small.

The image below illustrates the performance when utilizing side view input Roop images to generate images.

Image used for roop

Side view guy

Side view girl

Front view guy

Front view girl

Figure 34: Generate image using side view roop input

From the image generated, it is clear that the side view face generation for both the girl and the guy is quite accurate. However, when attempting to

generate a front view using a side face, the results are less precise, particularly in capturing the accurate shape of the face.

The image below illustrates the performance when utilizing font view input Roop images to generate images.

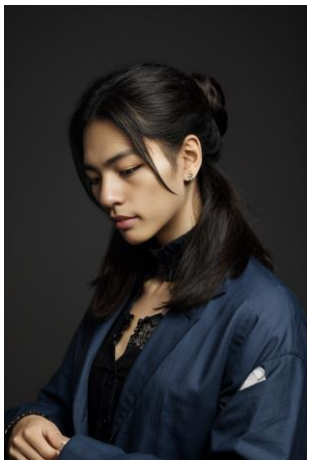
Image used for roop

Front view guy

Front view girl

Side view guy

Side view girl



From the image above, it's evident that the front view image generated from the side view is better than using the side view directly. The features of the person are mostly captured, including the slight protrusion of the forehead

to eyebrow area. However, the shape of the face is not entirely accurate. Conversely, the side view generated image is quite accurate.

Based on the above experiment, it is preferable to use a clear front view as a reference image for face replacement.

3.4 Generating images using all the provided tips

In this chapter, we have traversed through all the experiments and tips learned thus far. Now, we are poised to apply these accumulated insights to generate images. Each image will be accompanied by its respective prompt and settings, providing a comprehensive overview of the image generation process. Through this consolidated approach, we aim to harness the collective wisdom gleaned from our experiments and tips to produce compelling and diverse imagery. Let us proceed to witness the results of this integrated approach.

Model: realvisxlV40_v40LightningBakedvae

Prmopt: epic fantasy rpg art, oil painting, book cover illustration, an male sorceror in (black flowing wizard robes embroidered with runes), mage, wearing an amulet necklace with a red gemstone,fine details, leaning forward, turning to the side, magic energy, (dark, deep shadows), dramatic, high details, detailed face, glowing piercing eyes, off-centered composition, wielding cosmic energy, casting a magic spell with arms raised, intricate gestures, dynamic pose, magical mist, glowing gaze, unleash power, Magical, fantastical, highly detailed, vivid, rich details, highlights, intense, enhanced contrast, cool desaturated colors, Cinematic, Cinematic action Shot,award-winning, professional, highly detailed, impressionism

Negative Prompt: deformed pupils, wristwatch, bindi, earrings, bare chest, aidxlv05_neg, sci-fi, modern, urban, futuristic, low detailed, bad anatomy, bad illustration, bad proportions

Sampling method : DPM++ 2M Karras

Steps: 30

Size: 832 X 1216

CFG: 4

Figure 35: Generated Image 1

Model: cyberrealistic_v40

Prmopt: guy sitting on a small hill looking at night sky, fflix_dmatter, back view, distant exploding moon, nights darkness, intricate circuits and sensors, photographic realism style, detailed textures, peacefulness, mysterious.

Negative Prompt: worst quality, blurry, low quality, lowres, watermark, out of frame, overly saturated, disfigured, ugly, bad, overexposed, underexposed, cropped, jpeg artifacts

Sampling method : DPM++ 2M Heun Karras

Steps: 24

Size: 512 X 768

CFG: 6

Figure 37: Generated image 2

Model: tfvSDXLBAKED_tfvSDXLV3BAKED

Prmopt: cyberpunk style, (intricately detailed cyber space neon tunnel background:1.3), a young man in cyberspace, hands in his pockets, (kaleidoscopic digital geometry, virtual realities, neon abstract cubic surrealism, digital universe, binary, glowing computer code:1.4), vast digital cyberscape, (wearing normal tattered dirty cloths, hoodie:1.5) <lora:detailed_notrigger:2>

Neative prompt: child, childish, canvas frame, (high contrast:1.2), (over saturated:2), ((disfigured)), ((bad art)), ((deformed)),((extra limbs)),((close up)),((b&w)), blurry, extra limbs, extra legs, extra arms, disfigured, deformed, cross-eye, body out of frame, blurry, bad art, bad anatomy, 3d render, poorly drawn hands, poorly drawn feet, poorly drawn face, out of frame,

Sampling method : DPM++ 2M Karras

Steps: 42

Size: 832 X 1216

CFG: 4

Figure 36: Generated image 3

Chapter 4: Design and Implementation

This chapter focuses on the practical implementation of PhotoCraft, our custom Web UI designed to interact with the image generation system. We'll delve into the design considerations, the technologies employed, and detail the development process.

4.1 Design Considerations

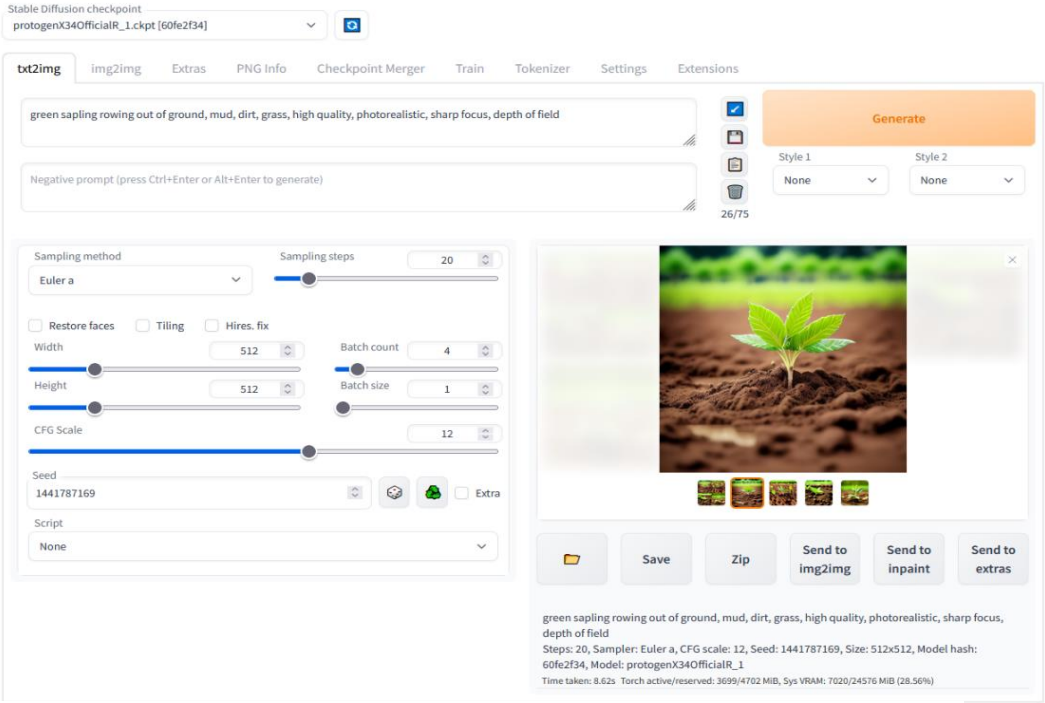


Figure 38: AUTOMATIC 1111 Web UI [7]

The original web UI, AUTOMATIC1111, is often considered too complex for beginners due to its multitude of settings and tabs, including the need to select models, install extensions like ControlNet and Roop manually, and configure sampling methods and steps to achieve high-quality images. This complexity may overwhelm novice users, particularly those unfamiliar with image generation techniques. Additionally, the extensive manual setup required for installing extensions may deter users from exploring advanced features.

Given that the primary user base comprises photographers who value realism in their images, it is imperative to streamline the user experience and make image generation more accessible. Therefore, the following design considerations are proposed for a more user-friendly experience:

I. User-Friendly Interface: Simplify the interface to make it more intuitive and accessible for beginners, reducing the need for extensive configuration.

II. Realistic Default Settings: Ensure that default settings for image generation prioritize realism, allowing users to achieve satisfying results without extensive adjustments.

III. Built-in Extensions: Integrate popular extensions like ControlNet and Roop directly into the platform, eliminating the need for manual installation. This allows users to access advanced features seamlessly without the hassle of setup. However, it is important to note that Roop

is currently not supported by the Stable Diffusion pipeline. Users will have to wait for future updates to enable Roop functionality.

IV. Incorporate Quick Guide: Include a comprehensive quick guide or tutorial section within the interface, offering step-by-step instructions and tips for users to get started quickly and effectively.

V. Accessibility with Low GPU Resources: Optimize the web UI to be compatible with lower GPU resources, ensuring that users with limited hardware capabilities can still utilize the platform effectively for image generation.

By implementing these design considerations, the web UI can become more user-friendly and accessible to beginners while still offering advanced features for experienced users, ultimately enhancing the overall user experience.

4.2 Technologies Utilized

I. React Frontend using Node.js: The frontend of the web application is built using React framework with Node.js for server-side rendering and enhanced developer experience.

II. Python Backend: The backend of the application is developed using Python programming language to handle server-side logic and data processing.

III. FastAPI: FastAPI is utilized as the middleware to connect the frontend and backend, providing efficient communication and routing between the two layers.

IV. Diffuser Library: The Diffuser library is integrated to leverage the Stable Diffusion pipeline for image generation, enabling advanced image processing capabilities within the application.

V. GitHub: GitHub is used for version control and collaborative development, facilitating for code management.

VI. Visual Studio Code (VSCode): VSCode is employed as the primary integrated development environment (IDE) for coding, providing a versatile and customizable environment for development tasks.

4.3 Development Process

The layout below introduces the user interface (PhotoCraft), designed specifically for photographers or new users. Throughout this session, we will meticulously explore each functionality of the web UI design ideas, addressing them in order from the top of the UI to the bottom.

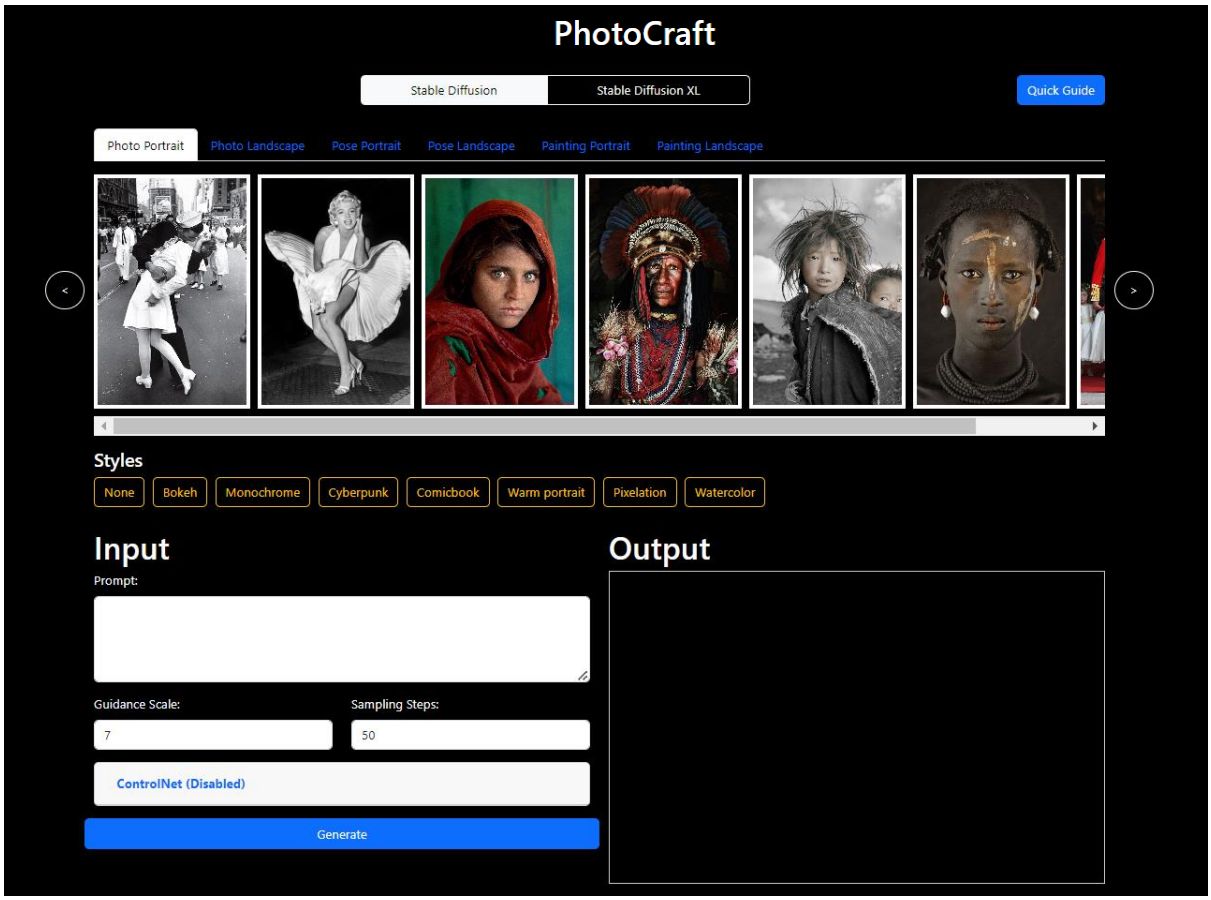


Figure 39: PhotoCraft UI

4.3.1 Model Selection

Figure 40: Model Selection

In the PhotoCraft interface, users cannot freely choose models but are limited to selecting either the SD1.5 or SDXL model. Different pipelines are employed based on the user's selection, utilizing the following four pipelines for image generation:

- StableDiffusionPipeline: Utilized when selecting SD1.5, with ControlNet disabled.
- StableDiffusionXLPipeline: Utilized when selecting SDXL, with ControlNet disabled.
- StableDiffusionControlNetPipeline: Utilized when selecting SD1.5, with ControlNet enabled.
- StableDiffusionXLControlNetPipeline: Utilized when selecting SDXL, with ControlNet enabled.

For the SD1.5 model, the Realistic_Vision_V5.1_noVAE model is utilized, while the RealVisXL_V4.0 model is chosen for SDXL. This setup allows users to experience both SD1.5 and SDXL models, catering to users with varying system resources.

4.3.2 Quick Guide

Figure 41: Quick Guide button

The "Quick Guide" button is conveniently located at the top right corner of the PhotoCraft UI. This guide provides users with a step-by-step walkthrough on how to utilize PhotoCraft effectively. It covers various actions, including selecting pipelines, adjusting settings, choosing styles, and increasing the weight in the prompt. While these steps are optional for image generation,

following them can greatly enhance the user experience. For detailed instructions, please refer to the Quick Guide located in Appendix A.

4.3.3 Image Banner

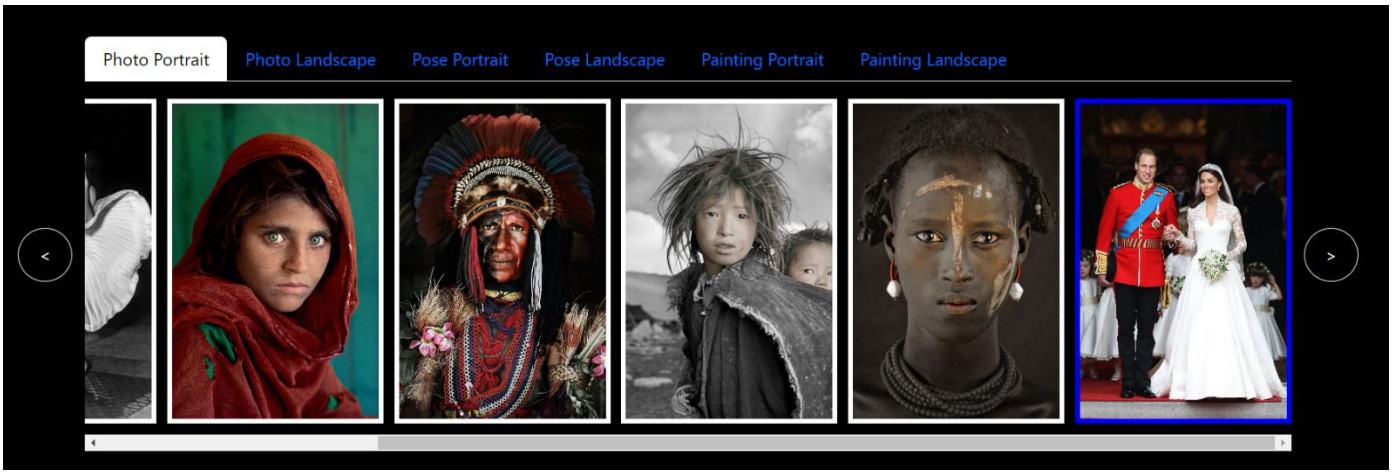


Figure 42: Image Banner

Famous landscape and portrait images, magazine covers, paintings, and poses are categorized and displayed on the image banner. When a user selects one of these images, it is boxed in blue, indicating selection. There are arrows on left and right of the banner for user to scroll left and right to select the image. Additionally, the ControlNet function is enabled, utilizing the selected image as a reference. However, this feature is optional for users, and they are not required to use it.

4.3.4 Styles

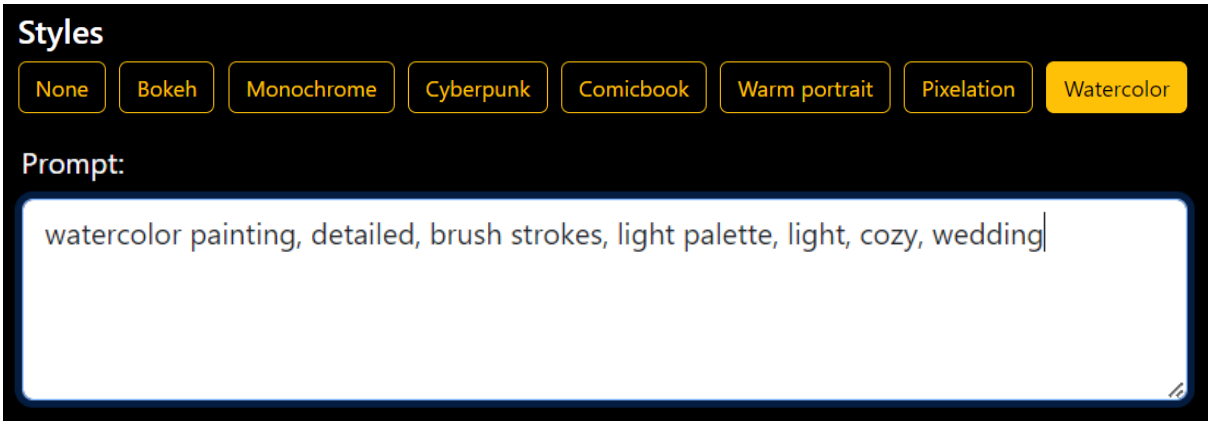


Figure 43: Prompt display according to the style

When a style is selected, PhotoCraft automatically generates words relevant to the prompt. This not only adds convenience for users but also provides an opportunity to educate beginners on how to write prompts effectively. Users can also add their own text to the prompt, allowing for further customization and personalization.

PhotoCraft does not require a database; instead, all data related to styles is stored in an Excel file. Whenever the Excel file is updated, including additions or removals of styles, the number of styles and their descriptions within PhotoCraft are automatically updated accordingly. The code is designed to handle these updates seamlessly.

	A	B
1	Style	Description
2	None	
3	Bokeh	Bokeh background
4	Monochrome	monochrome, high contrast, dramatic shadows, 1940s style, mysterious, cinematic
5	Cyberpunk	a glamorous digital magazine photoshoot, a fashionable model wearing avant-garde clothing, set in a futuristic cyberpunk roof-top environment, with a neon-lit city background, intricate high fashion details, backlit by vibrant city glow, Vogue fashion photography

Figure 44: Excel file for storing styles

4.3.5 Default setting

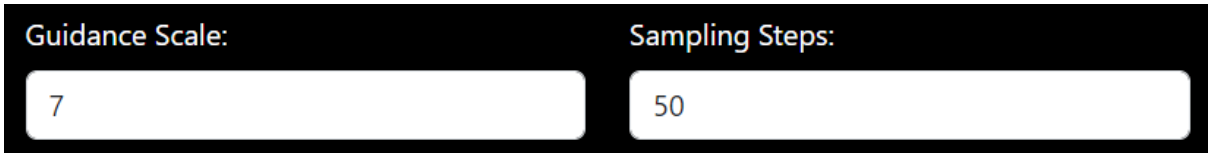


Figure 45: Figure 44: Default Value of CFG and Steps

The default value for Guidance Scale (CFG) is set to 7, and the sampling steps are set to 50. Users have the freedom to adjust these parameters, but they are constrained within specific ranges:

Guidance Scale ranges from 1 to 20.
Sampling steps range from 1 to 150.

If users input values outside of these ranges, the UI will prompt an error message, notifying them to select a correct value within the specified range. The following images illustrate the error message displayed when the input number is not within the valid range.

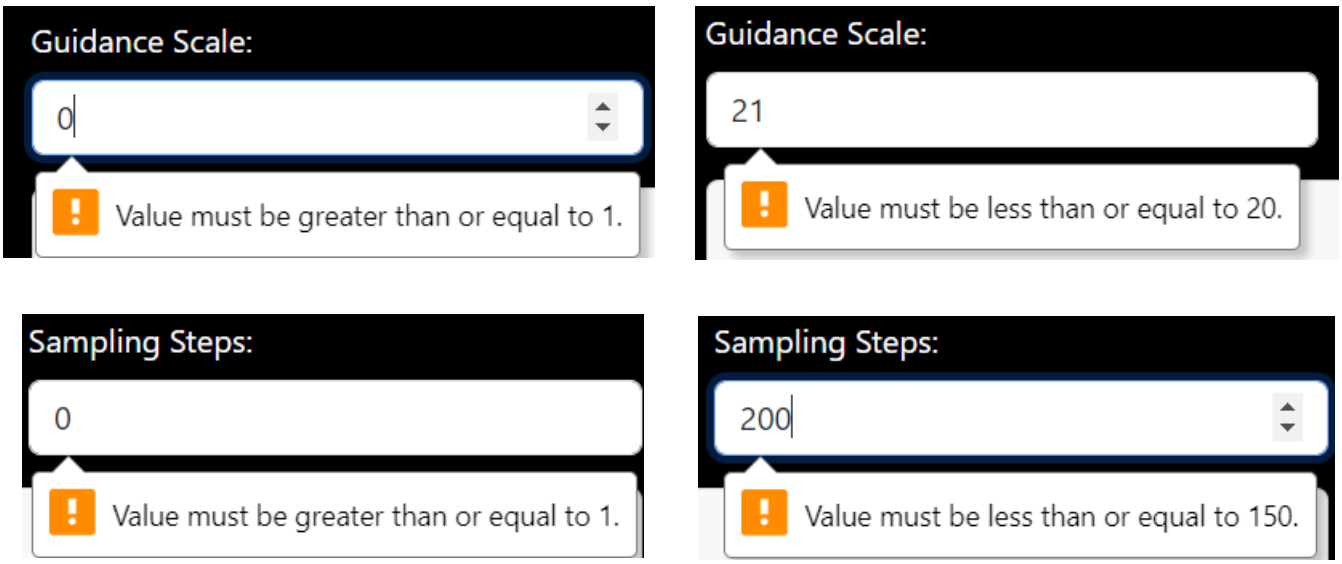


Figure 46: Errors when number out of range

Negative Prompt:

The universal negative prompt, as discussed in session 3.1.2 Experiment 4, is utilized in the background of the UI therefore it is not included in the UI interface for users to interact with directly.

4.3.6 ControlNet

ControlNet has been effectively integrated for Canny, OpenPose, and Depth in the SD1.5 model. However, for the SDXL model, ControlNet is available only for Canny and Depth, as OpenPose support is currently not available for SDXL. Other ControlNets are not available for the current pipelines.

The ControlNet card is expandable, meaning it can be toggled open or closed. When ControlNet is disabled, the card remains closed. However, when ControlNet is enabled, the card expands automatically. Users can enable or disable ControlNet by clicking on the ControlNet card. If an image from the Image Banner is selected, ControlNet will be automatically enabled, and the selected image will be displayed alongside the ControlNet image. Additionally, users can choose their own images by clicking on the "Choose File" button. The green button in the ControlNet card is a dropdown list where users can select which ControlNet to use. The default ControlNet is Canny.

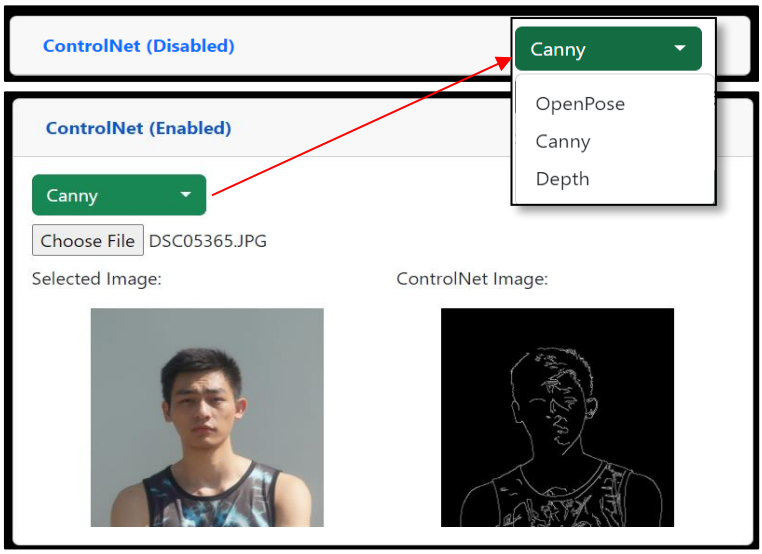


Figure 47: Disable/Enable ControlNet

4.4 Project Directory Structure

In this section, we'll provide a general overview of how to organize and manage the codebase effectively.

In the project directory, we have a clear separation between the backend (API) and frontend (client) components.

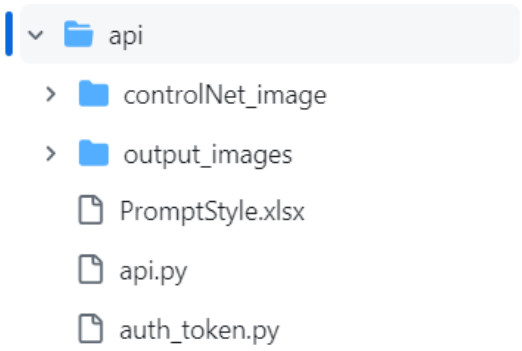


Figure 48: Backend structure

Backend (API) Directory Structure:

- ControlNet_image: This directory is dedicated to storing ControlNet images. Each image here will be replaced once a new ControlNet image is generated.
- output_images: All generated images are stored here, organized by date and time to ensure uniqueness and prevent overwriting.
- PromptStyle.xlsx: This file contains prompt styles and is used for storing prompt styles.
- api.py: The main Python file where backend code is written.
- auth_token.py: This file is used to store the passcode required for accessing Hugging Face pipelines.

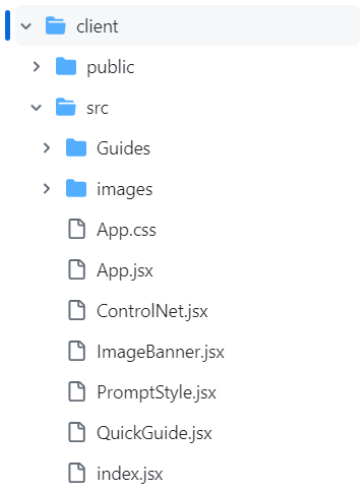


Figure 49: Frontend structure

Frontend (Client) Directory Structure:

- public: This directory, installed by Node.js, contains all the libraries of Node.js.
- guide_images: Images used in the quick guides are stored here.
- images: All images used in the image banner are stored in this directory.
- index.jsx: The main JavaScript file where frontend code is written. This file serves as the entry point for the client-side application.
- Other .jsx files: Additional JavaScript files containing components that help in maintaining code readability and organization.

This directory structure helps keep backend and frontend components separate, making it easier to manage and maintain each part of the application. Additionally, it provides clear organization for different types of files and resources used in the project.

4.5 installations Guide

The PhotoCraft code is available on GitHub at the following repository: https://github.com/JX0830/PhotoCraft

Prerequisites:

I. Visual Studio Code (vsCode): Download and install vsCode.
II. Git: Download and install Git.
III. Python 3.10.6: Download and install Python 3.10.6 and add it to your system's PATH.
IV. Node.js: Download and install Node.js.

Installation Steps:

I. Clone PhotoCraft Repository: Clone the PhotoCraft repository to your local directory.
II. Install Dependencies: Open your command prompt and install the following Python dependencies:
    - pip install uvicorn
    - pip install pandas
    - pip install openpyxl
    - pip install fastapi
    - pip install diffusers
    - pip install opencv-python
    - pip install controlnet-aux==0.0.3
    - pip install transformers
    - pip install python-multipart

- pip install accelerate
- pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118

III. Running the API Server:

- Use VSCode to open the cloned PhotoCraft repository.
- Right-click on the api directory and select "Open in Integrated Terminal".
- In the terminal, execute the following command:
  uvicorn api:app --reload

IV. Running the Client Server:

- Right-click on the client directory and select "Open in Integrated Terminal".
- Run the following command to install the necessary dependencies:
  npm install
- After installation, start the development server by running:
  npm start

4.6 Testing

Testing is crucial in software development, and currently, only gray box testing is being utilized due to its numerous advantages. Gray box testing, which blends elements of both black box and white box testing, offers a balanced and effective approach to testing software systems. Please refer to Appendix B for the gray box testing table.

4.7 Limitations

I. The latest extensions and features available on AUTOMATIC1111 may not be immediately usable in the pipeline for PhotoCraft. Developers may need to wait until the pipelines are updated to incorporate these features into PhotoCraft. For example, the Roop extension is currently unavailable for the pipelines.

II. The pipelines currently support only models found on the Hugging Face website and do not support models from other platforms.

III. The exclusive reliance on gray box testing as the primary testing methodology, while ensuring a thorough examination of system functionality and implementation, may potentially overlook issues that could be identified through other testing approaches such as user testing and system testing.

Chapter 5: Conclusion

1. Conclusion on Exploration of Stable Diffusion Parameters and Extensions

The exploration of Stable Diffusion parameters and extensions has shed light on the vast potential for high-quality portrait image generation. By delving into the nuances of parameter optimization, users can refine their image generation processes with precision and efficiency. The integration of ControlNet and Roop extensions further enriches the experience by providing enhanced control and flexibility over image composition and facial features.

Additionally, the diverse range of face replacement techniques offers users versatile options for seamless image manipulation. Whether through retraining models, inpainting, or leveraging the Roop extension, digital artists and photographers have valuable tools at their disposal to enhance or transform their images.

As Stable Diffusion continues to evolve through ongoing development and refinement, with the introduction of new extensions, it holds the potential to become a staple tool in the toolkit of creative professionals and enthusiasts alike. Each extension developed adds to the platform's capabilities, expanding the horizons of image generation and manipulation possibilities.

2. Conclusion on Customed UI PhotoCraft

Beyond the technical aspects, the PhotoCraft project represents a significant leap towards democratizing the Stable Diffusion framework for a broader audience. Through its user-friendly web interface, PhotoCraft simplifies the complex process of image generation, catering to users with varying resource capabilities. By incorporating essential features like ControlNet and providing a Quick Guide, PhotoCraft ensures that users can navigate the platform with confidence and ease.

While challenges may arise, such as adapting to updates in the Stable Diffusion pipeline, PhotoCraft remains a promising tool for creative expression in photography and digital art. As the pipeline evolves and updates are integrated, the potential for PhotoCraft to further expand its feature set and usability grows, promising an even more enriching experience for users in the realm of digital creativity.

Bibliography

[1] Guinness, H. (2024, February 22). The best AI image generators in 2024. Zapier. Retrieved from https://zapier.com/blog/best-ai-image-generator/

[2] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021, February 10). Score-based generative modeling through stochastic differential equations. Retrieved from https://arxiv.org/abs/2011.13456

[3] Andrew (2024, January 5). How does Stable Diffusion work? Stable Diffusion Art. Retrieved from https://stable-diffusion-art.com/how-stable-diffusion-work/

[4] Harry Nguyen (2024, February 21). What is stable diffusion and how does it work? TECHVIFY Software. Retrieved from https://techvify-software.com/what-is-stable-diffusion/

[5] Arnold, V. (2024, February 15). Understanding stable diffusion: Advantages and limitations. Neuroflash. Retrieved from https://neuroflash.com/blog/understanding-stable-diffusion-advantages-and-limitations/#:~:text=While%20stable%20diffusion%20has%20several,and%20the%20network%20parameters%20used.

[6] ANDY H. TU,(2024, March 18). SYSTEM REQUIREMENTS FOR STABLE DIFFUSION: YOUR COMPLETE GUIDE. Retrieved from https://andyhtu.com/system-requirements-your-complete-guide-to-running-stable-diffusion-efficiently/

[7] GitHub(2024, March 18). AUTOMATIC1111/stable-diffusion-webui. Retrieve from https://github.com/AUTOMATIC1111/stable-diffusion-webui

[8] Guide to stable diffusion Prompt Weights | Getimg.ai. (n.d.). https://getimg.ai/guides/guide-to-stable-diffusion-prompt-weights

[9] Andrew. (2023, June 10). Stable Diffusion Samplers: A Comprehensive guide - Stable Diffusion Art. Stable Diffusion Art. https://stable-diffusion-art.com/samplers/

[10] (2024, March 18) Guide to Stable Diffusion CFG scale (guidance scale) parameter | getimg.ai. (n.d.). https://getimg.ai/guides/interactive-guide-to-stable-diffusion-guidance-scale-parameter

Appendices A: Quick Guide on Using PhotoCraft

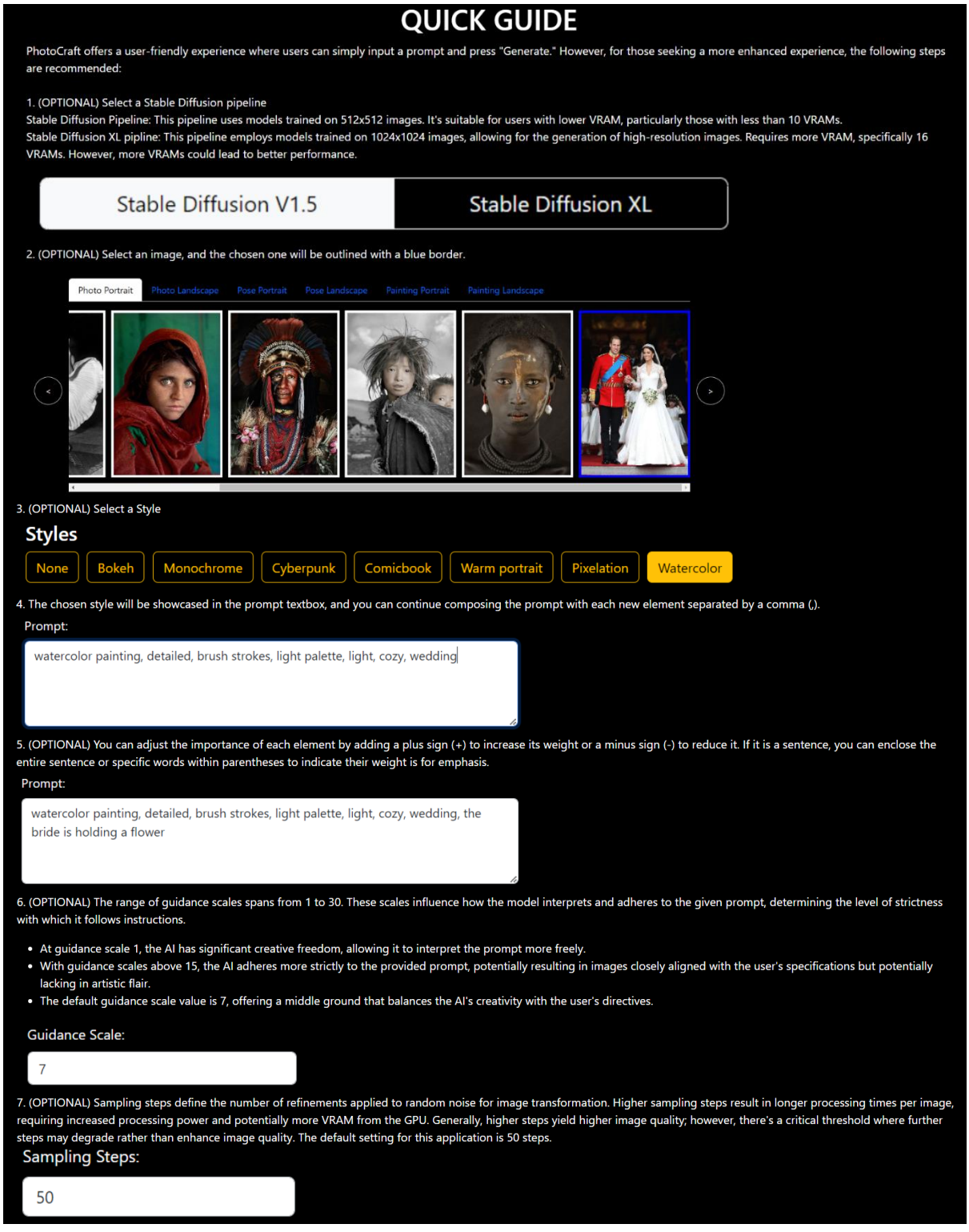


Figure 50: Quick Guide Part 1

Figure 51: Quick Guide Part 2

Appendices B: Gray Box Testing

Test Case	Description	Input	Expected Output	Actual Output	Status
1	Select Stable Diffusion XL	Click on Stable Diffusion XL toggle button	Stable Diffusion XL is highlight in white ControlNet drop box only shows Canny and Depth available	Stable Diffusion XL is highlight in white ControlNet drop box only shows Canny and Depth available	Pass
2	Select Stable Diffusion V1.5	Click on Stable Diffusion V.5 toggle button	Stable Diffusion XL is highlight in white ControlNet drop box only Canny, OpenPose and Depth available	Stable Diffusion XL is highlight in white ControlNet drop box only shows Canny, OpenPose and Depth available	Pass
3	Viewing Quick Guide	Click on Quick Guide Button	Open a Quick Guide page	Open a Quick Guide page	Pass
4	Select Image on Image Banner	Click on a random image on image banner	Clicked image bordered in blue, ControlNet enabled with Canny image shown	Clicked image bordered in blue, ControlNet enabled with Canny image shown	Pass
5	Test Image Banner Scroller	Click left and right of the scroller	Image banner scrolls left/right respectively	Image banner scrolls left/right respectively	Pass
6	Select Monochrome Style	Click on Monochrome Style	Prompts: “monochrome, high contrast, dramatic shadows, 1940s style, mysterious, cinematic” displayed	Prompts: “monochrome, high contrast, dramatic shadows, 1940s style, mysterious, cinematic” displayed	Pass
7	Set Guidance Scale (Lower Limit)	Input: -10	Error: Value must be greater than or equal to 1	Error: Value must be greater than or equal to 1	Pass
8	Set Guidance Scale (Upper Limit)	Input: 50	Error: Value must be less than or equal to 20	Error: Value must be less than or equal to 20	Pass

9	Set Sampling Step (Lower Limit)	Input: -10	Error: Value must be greater than or equal to 1	Error: Value must be greater than or equal to 1	Pass
10	Set Sampling Step (Upper Limit)	Input: 200	Error: Value must be less than or equal to 150	Error: Value must be less than or equal to 150	Pass
11	Expand ControlNet Card	Click on ControlNet Closed card	ControlNet card expands	ControlNet card expands	Pass
12	Collapse ControlNet Card	Click on ControlNet Open card	ControlNet card closes	ControlNet card closes	Pass
13	Generate Image	Write a prompt, then click on Generate image	Generated image resembles the description	Generated image resembles the description	Pass
14	Use Canny ControlNet to Generate Image	Choose an image, select Canny on ControlNet dropdown, click Generate button	Selected image and Depth ControlNet image displayed, generated image resembles selected image	Selected image and Depth ControlNet image displayed, generated image resembles selected image	Pass
15	Use Depth ControlNet to Generate Image	Choose an image, select Depth on ControlNet dropdown, click Generate button	Selected image and Depth ControlNet image displayed, generated image resembles selected image	Selected image and Depth ControlNet image displayed, generated image resembles selected image	Pass
16	Use OpenPose ControlNet to Generate Image	Choose an image, select OpenPose on ControlNet dropdown, click Generate button	Selected image and OpenPose ControlNet image displayed, generated image resembles selected image	Selected image and OpenPose ControlNet image displayed, generated image resembles selected image	Pass

Table 5: Gray Box Testing