Img2Img-Controlnet-ComfyUI

Introduction:

This project leverages Controlnet within the ComfyUI framework to explore two artistic styles: GTA and Anime. The goal is to produce high-quality images while maintaining efficiency and image quality.

Workflow:

The workflow for the Img2Img-Controlnet-ComfyUI project consists of several stages, each crucial for the overall performance of the system. The key stages are as follows:

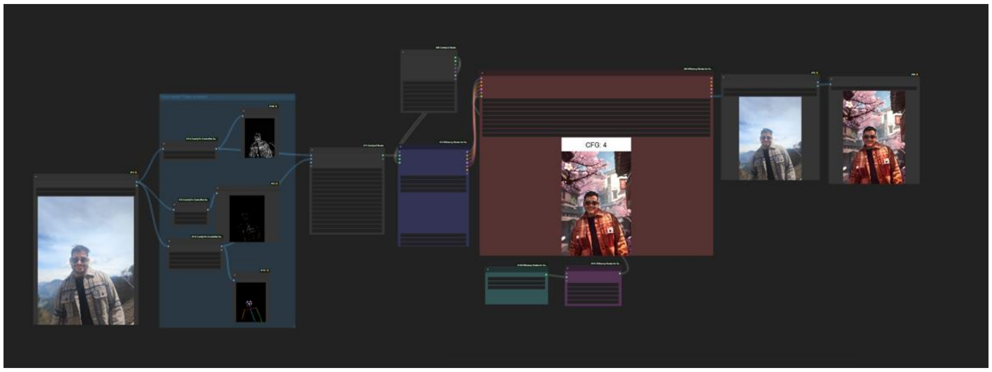
1. Input Image: The process begins by passing the input image through LineArt and OpenPose preprocessors.

2. ControlNet: The preprocessed images are fed into ControlNet for further processing.

3. Efficient Loader: ControlNet outputs are then passed to the Efficient Loader, responsible for loading the necessary weights.

4. KSampler: Finally, the loaded data is processed through KSampler to generate the output image.

The process is depicted in the workflow diagram (Figure 1).



Custom Nodes:

The project utilizes several custom nodes to enhance functionality and performance. These custom nodes are divided into three categories: Comfyroll Studio, ComfyUI, and ComfyUI Nodes for Inference.Core.

The Comfyroll Studio category includes two custom nodes: CR Aspect Ratio and CR Multi-ControlNet Stack. The CR Aspect Ratio node adjusts the aspect ratio of images, while the CR Multi-ControlNet Stack node manages multiple ControlNet instances.

The ComfyUI category includes three custom nodes: PreviewImage, SaveImage, and LoadImage. The PreviewImage node previews images during processing, the SaveImage node saves the generated images, and the LoadImage node loads images for processing.

The ComfyUI Nodes for Inference.Core category includes three custom nodes: CannyEdgePreprocessor, OpenposePreprocessor, and LineArtPreprocessor. The CannyEdgePreprocessor node preprocesses images using canny edge detection, the OpenposePreprocessor node uses OpenPose for human pose estimation, and the LineArtPreprocessor node converts images to line art for further processing.

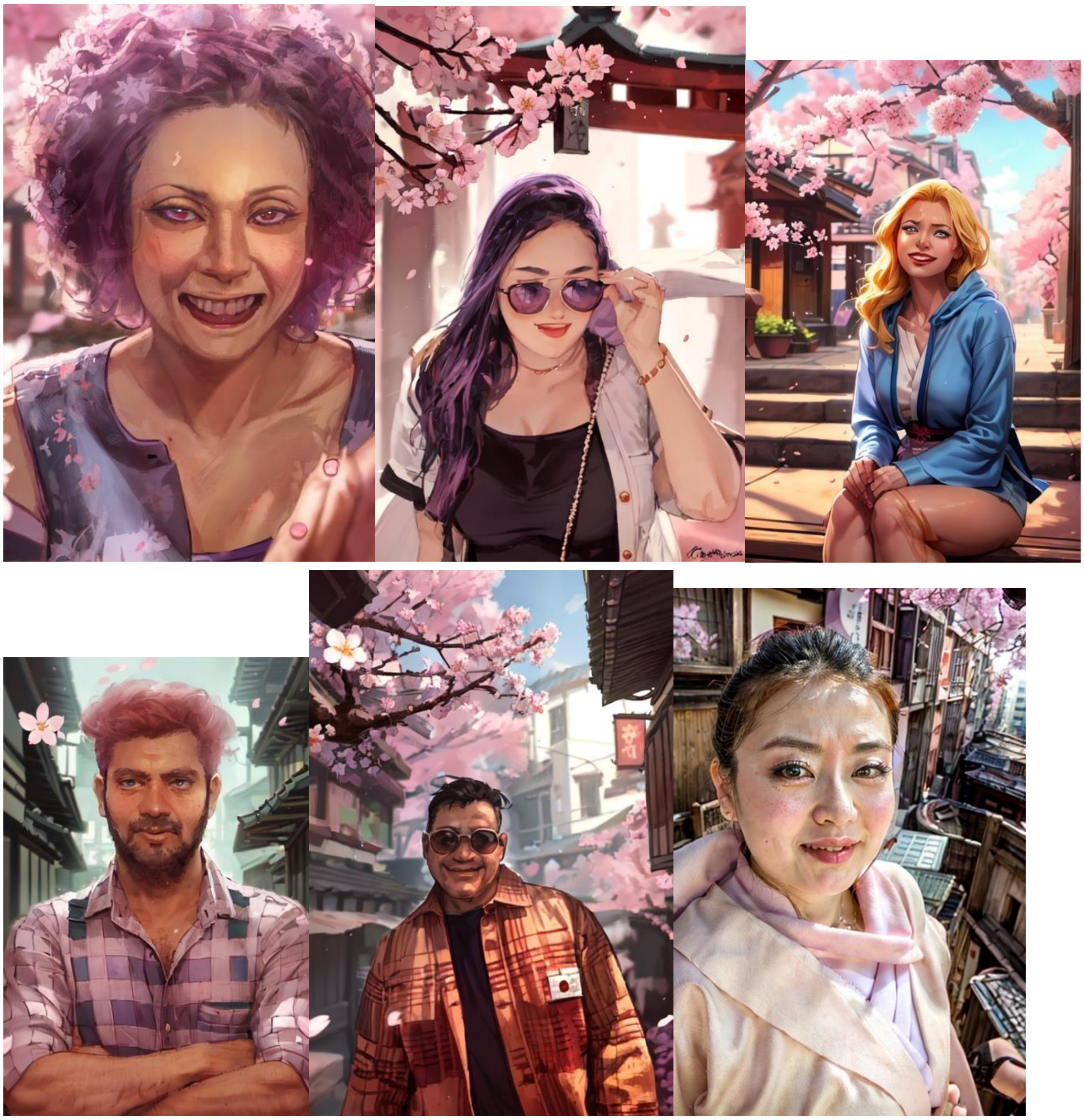
Models - Checkpoint and VAE:

The project uses specific checkpoints and VAE models to achieve the desired output quality. The checkpoint used is the Realistic Vision V6.0, while the VAE model is the SD VAE. Finding the correct balance between checkpoints and weights for both checkpoints and VAE was challenging, as it is crucial to ensure high-quality output without compromising efficiency.

Output Images - GTA :

-

Output Images – Anime:



Docker:

To facilitate easy setup and deployment of the project, a Docker file is included in the repository. This ensures a consistent and reproducible environment for running the Img2Img-Controlnet-ComfyUI system.

Discussion:

Despite the challenges in balancing checkpoints and weights, the current setup provides a robust framework for image-to-image translation in the specified styles. The GTA version performed better than the anime version in generating cleaner and faster outputs.

Future Considerations:

Future work aims to make the model more robust and capable of generating cleaner and faster outputs, building on the success of the GTA version. This involves fine-tuning the existing models and exploring new techniques to enhance performance and image quality.

Conclusion:

The Img2Img-Controlnet-ComfyUI project showcases a comprehensive approach to image-to-image translation using Controlnet on ComfyUI. While challenges in balancing checkpoints and weights exist, the project provides a solid foundation for further research and development, with the GTA version demonstrating promising results.