Investigating a Resource-Efficient Approach to Photo-realistic Image Generation from Sketches using Stable Diffusion

Ramsha Perwez
College of Computing & Informatics
Drexel University
Philadelphia, PA 19104, USA
rp955@drexel.edu

Abstract

Pre-trained models are useful for challenging tasks that require machine learning and deep learning. This project explores transforming simple sketches into high-resolution images using Convolutional Neural Networks (CNN), Stable Diffusion and Generative Adversarial Networks (GANs). Initial sketches are pulled in from the QuickDraw dataset and processed with CNN. After the initial pass, the sketches are passed through the Stable Diffusion model to generate a preliminary image. Essential libraries like diffusers, transformers, and torch are installed to manage image processing, machine learning models, and hardware acceleration. There are numerous pre-trained models available, which can be fine-tuned for specific use cases, such as generating photorealistic images. This project uses the RunwayML Stable Diffusion model which generates a high-quality image corresponding to the sketch and a text prompt. This project explores various approaches and evaluates the results by comparing the generated images to the original sketches and assess for image quality.

1 Introduction

The generation of high-quality images from simple sketches holds immense potential across various fields, from design and law enforcement to education and accessibility. This project leverages the power of deep learning models. Convolutional Neural Networks excel at producing visual patterns, crucial for interpreting sketches. Stable Diffusion, a text-to-image model refines images based on textual guidance, providing flexibility and control [1]. This project specifically focuses on the combination of ControlNet and Stable Diffusion to achieve the desired image transformation, highlighting their complementary strengths in bridging the gap between sketches and photorealistic visuals.

2 Utilizing Quick, Draw!

Quick, Draw! is a popular dataset developed by Google in which everyday users can participate and train its neural network model. The platform invites users to draw various objects and concepts within a limited time frame, meanwhile attempting to recognize the drawings using machine learning. This interactive tool has led to an extensive data set which contain millions of sketches across numerous categories. In this project, Quick, Draw! serves as the source of initial sketches that are transformed into high-resolution images, demonstrating the practical applications of not only this dataset but other datasets in which everyday users are able to contribute for more advanced topics [6].

3 Runway ML Stable Diffusion V1.5

Since this project is experimenting with an available data set of sketches, using a model that specializes in enhancing sketch images is key. Stable diffusion is generative model that transforms a 'noisy' image to a refined result. The model used is a version of Stable Diffusion released by RunwayML [8]. The provided code generates photorealistic images of the chosen category based on sketches provided from Quick, Draw. It's important to note that this model doesn't directly "use text" in the traditional sense. Instead, it's a text-to-image model, meaning it uses text input (prompts) as guidance to generate images. It is worthy to note that the specific model used is compatible with ControlNet, which can be useful for this experiment as it leverages the sketch as a guiding input for image generation. The generated image is then optionally upscaled using basic bicubic interpolation for enhanced detail and clarity. The code is optimized for both GPU and CPU execution and includes measures to manage memory usage, making it suitable for environments with limited resources [7].

3.1 Img2Img Pipeline with Stable Diffusion

An option that works well with sketch to image is the Img2Image model, which is a specialized application of stable diffusion inheriting aspects from DiffusionPipeline. There are multiple parameters that can be adjusted to alter the image quality such as strength, guidance scale as well as number of inference steps. Altering the strength will control the deviation from the original input. Similarly, adjusting the number of inference steps will control the steps of the diffusion itself. The benefit of the adjustments will result in a higher quality image but will take significant longer for processing [5].

4 Methodology

After the necessary software libraries are installed and the Stable Diffusion v1-5 mode is initialized, this experiment following main steps. First, a sketch is generated based on the selected category using Google's Quick, Draw! dataset. This sketch is then passed to the stable diffusion model and an enhancer which generates a detailed image.

4.1 Functions

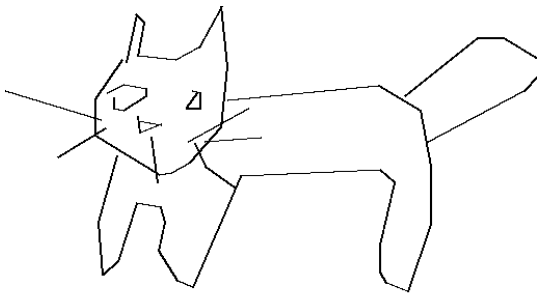
The `get_sketch` function retrieves a random sketch from the dataset, handling potential issues like extreme aspect ratios and placing the sketch on a white background for consistency. The `get_image_from_sketch` function uses the fine-tuned Stable Diffusion model. It takes the sketch and a detailed text prompt as input, guiding the model to generate a high-resolution photorealistic image. Parameters such as `strength` and `guidance_scale` are adjusted to control the level of detail and adherence to the original sketch.

4.2 Image Parameters

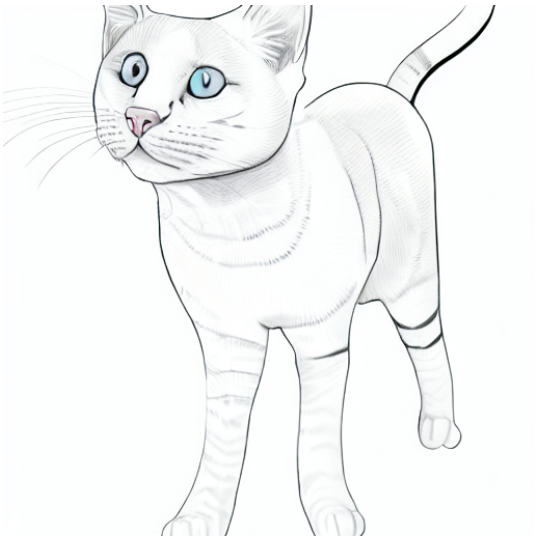
This sketch is scaled to fit a 512x512 pixel canvas, ensuring compatibility with the model's input requirements. The model uses carefully tuned parameters, including a strength of 0.75 and a guidance scale of 7.5, to balance adherence to the input sketch with the creative interpretation of the prompt. The strength parameter, set at 0.75, acts as a weighting factor that determines how closely the generated image adheres to the sketch. A higher strength value results in an image that closely mirrors the sketch's lines and shapes, while a lower value allows for more abstraction. The guidance scale directly controls how much the final image resembles the sketch. Higher values force the image to closely follow the sketch's lines, while lower values allow for more creative interpretations, potentially adding new elements not present in the original drawing [2].

5 Visual Image Assessment

The visual assessment of the generated image reveals a successful enhancement that aligns well with the original sketch and the prompt. The model has captured the essence of the cat's pose and structure, showcasing its photorealistic potential. However, the image lacks vibrancy, presenting a somewhat muted color palette. Furthermore, the upscaling process did not yield any improvement in visual quality. Potential reasons for the lack of color variation could be attributed to the nature of the sketch itself, which have been primarily grayscale. The lack of noticeable improvement after upscaling may be because the initial image was already high-resolution or the upscaling technique wasn't suited to this image type (sketches).

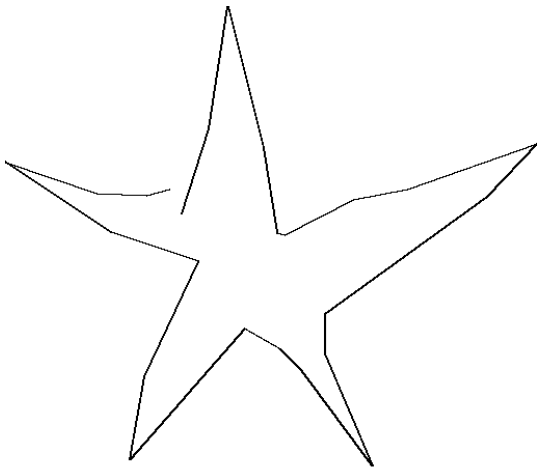


(a) Initial Sketch of Category: "Cat" retrieved from Quick, Draw!

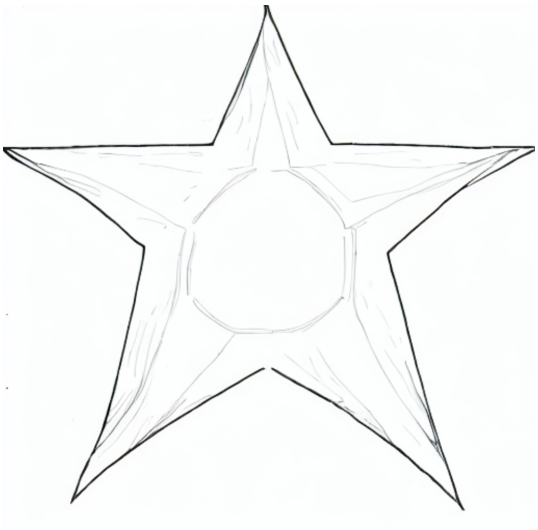


(b) Enhancement with Stable Diffusion

Figure 1: Comparison of Initial Sketch and Enhanced Image



(a) Initial Sketch of Category - "Star" retrieved from Quick, Draw!



(b) Enhancement with Stable Diffusion

Figure 2: Comparison of Initial Sketch and Enhanced Image for the Category "Star"

The following Category - "table" compares between the initial RunwayML model to an upscaled resolution. As explained, there are no direct visual differences.

(a) RunwayML Implementation

(b) Enhanced Resolution

Figure 3: Comparison between RunwayML Implementation and Enhanced Resolution for the Category "Table"

6 Exploring Generative Adversial Networks

6.1 The Option of ESGRAN and Real-ESGRAN

ESRGAN and its derived Real-ESRGAN are image super-resolution models that excel at enhancing the quality of low-resolution images. Although other choices such as ResNet are available, ESRGAN models are specifically designed to restore details of poor quality images. It leverages a high-order degradation model to mimic real-world image imperfections like blurring and noise, enabling it to better restore realistic details. In this project, Real-ESRGAN was employed to refine the initial images generated by Stable Diffusion from sketches. Given that Real-ESRGAN relies on older PyTorch functionalities, there were extensive adjustments made to the code slightly to align with newer versions. Specifically, there was a need to modify the import statements within the `realesrgan_dataset.py` and `degradations.py` files to ensure compatibility with the newer `torchvision.transforms.functional` module instead of the deprecated `torchvision.transforms.functional_tensor`. This modification allows Real-ESRGAN to seamlessly integrate into the pipeline, ultimately producing higher-quality images [3].

6.2 Adjustment to the architecture of the Real-ESRGAN model and the pre-trained weights model

When loading the pre-trained weights into the Real-ESRGAN model, the code can generate a Runtime Error. This is because the model architecture defined in the code might not match the structure of the pre-trained weights. Specifically, the code was creating a model with fewer layers than the pre-trained model expected. This mismatch prevented the weights from loading correctly. The solution is to adjust the model's initialization parameters, increasing the number of convolution layers (`num_conv`) from 16 to 32. This change ensures that the model structure matched the pre-trained weights, allowing for successful loading. This highlights the importance of aligning model architectures when working with pre-trained weights in deep learning projects.

7 Conclusion

This project demonstrated successful image generation from sketches using ControlNet and Stable Diffusion, highlighting their potential in creative applications. The current limitations in color vibrancy and upscaling point to future enhancements, possibly through the integration of additional models for

colorization and advanced upscaling techniques. Future research could explore combining multiple AI models to achieve even more impressive results. The project also underscores the significance of adapting and adjusting existing models and their parameters, like in the case of Real-ESRGAN, to achieve optimal results and compatibility with evolving software environments. This adaptability is crucial in the ever-progressing field of deep learning, ensuring that tools and techniques remain effective and relevant.

References

[1] Aristimuño, Ignacio. "An Introduction to Diffusion Models and Stable..." Marvik, 29 Nov. 2023. https://blog.marvik.ai/2023/11/28/an-introduction-to-diffusion-models-and-stable-diffusion.

[2] Ashvanth.S. "A Comprehensive Guide to Stable Diffusion Parameters for Image Generation." Automagically by Segmind, 30 May 2024. https://blog.segmind.com/a-comprehensive-guide-to-stable-diffusion-parameters-for-image-generation.

[3] Cochard, David. "Real ESRGAN: Super-Resolution Model Enhanced for Denoising." Medium, 11 Jan. 2024. https://medium.com/axinc-ai/real-esrgan-super-resolution-model-enhanced-for-denoising-dd581b2702a8.

[4] De Souza, Vinicius Luis Trevisan, et al. "A Review on Generative Adversarial Networks for Image Generation." Computers & Graphics, vol. 114, Aug. 2023, pp. 13–25. https://doi.org/10.1016/j.cag.2023.05.010.

[5] "Image-to-image." Hugging Face. https://huggingface.co/docs/diffusers/en/using-diffusers/img2img.

[6] Quick, Draw! quickdraw.withgoogle.com. "Quickdraw." PyPI, 16 May 2023. https://pypi.org/project/quickdraw.

[7] Rath, Sovit, and Sovit Rath. "ControlNet – Achieving Superior Image Generation Results — LearnOpenCV ." LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow With Code, & Tutorials, 28 Feb. 2024. https://learnopencv.com/controlnet.

[8] "Runwayml/Stable-diffusion-v1-5." Hugging Face. https://huggingface.co/runwayml/stable-diffusion-v1-5.