Training a ControlNet for Stable Diffusion

Project Progress Report - 6.8300

Hermine Tranie
Massachusetts Institute of Technology
hermitra@mit.edu

Abstract

This project aims to train a ControlNet, a deep learning algorithm, to control Stable Diffusion on a new condition. The ControlNet takes in a control image and a text prompt and produces a synthesized image that matches the prompt and follows the constraints imposed by the control image. This project proposes to train a new condition and qualitatively analyze the results in terms of prompt fidelity, condition fidelity, and quality of the resulting imagery. This project objectives are to train a ControlNet on a condition of my own, evaluate the performance of the trained ControlNet on a properly defined test set, and analyze the results and identify potential areas for improvement.

1. Introduction

This project focuses on the application and training of ControlNet, to manage Stable Diffusion under a novel condition. ControlNet has the ability to control image synthesis tasks by taking a control image and a text prompt as input, and subsequently generating a synthesized image that adheres to the prompt while conforming to the constraints set by the control image. By introducing a new condition, this project seeks to qualitatively assess the outcome in terms of prompt fidelity, condition fidelity, and the overall quality of the generated imagery.

1.1. ControlNet

ControlNet is an end-to-end neural network architecture designed for controlling large image diffusion models in task-specific input conditions. It utilizes a unique ”zero convolution” layer to connect trainable and locked copies of the model, preserving the model’s capabilities while learning conditional control efficiently. ControlNet demonstrates effectiveness across various conditions and datasets in image processing tasks.

1.2. Stable Diffusion

Stable diffusion is a deep learning approach for image synthesis that leverages diffusion models to generate images by reversing a denoising process. This method produces high-quality images while maintaining computational efficiency. It has been used in various applications such as text-to-image synthesis, inpainting, and class-conditional image generation.

1.3. Objectives

The core objectives of this project include training a ControlNet with a custom condition, evaluating its performance on a well-defined test set, and analyzing the results to pinpoint potential areas for enhancement and further development.

2. Related Work

2.1. High-Resolution Image Synthesis with Latent Diffusion Models [1]

The paper by Rombach et al. (2022) introduces Latent Diffusion Models (LDMs), an approach for high-quality image synthesis that significantly reduces computational requirements compared to traditional pixel-based diffusion models. By training diffusion models in the latent space of pretrained autoencoders and introducing cross-attention layers, LDMs achieve state-of-the-art performance in tasks such as image inpainting and class-conditional image synthesis, while maintaining competitive results in text-to-image synthesis, unconditional image generation, and super-resolution. The LDMs offer a scalable, efficient, and versatile solution for image synthesis and provide insights into the potential of latent space conditioning for diffusion models.

2.2. Adding Conditional Control to Text-to-Image Diffusion Models [2]

ControlNet, introduced in a recent paper, is an end-to-end neural network architecture that addresses challenges in controlling large image diffusion models for task-specific input conditions. Considering the limitations in data scale, computational resources, and diverse problem definitions in various image processing tasks, the authors develop a method that clones the weights of a diffusion model into a ”trainable copy” and a ”locked copy.” This approach maintains the network’s capabilities while enabling efficient learning of conditional control on task-specific datasets through a unique ”zero convolution” layer. The effectiveness of ControlNet is demonstrated across multiple conditions and datasets, such as Canny edges, Hough lines, user scribbles, human key points, segmentation maps, shape normals, and depths. Furthermore, the authors highlight that ControlNet can achieve competitive results with commercial models in tasks like depth-to-image, even when trained on personal computers with limited resources.

2.3. Building on related work

By building upon the insights from these two papers, our project seeks to explore the potential of ControlNet for controlling Stable Diffusion with a new condition and to analyze the performance of the trained ControlNet with respect to prompt fidelity, condition fidelity, and image quality.

3. Data

3.1. Datasets

For the first experiment, we are using the fill50k data available fron Hugging Face here: https://huggingface.co/lllyasviel/ControlNet/blob/main/training/fill50k.zip. This dataset contains as source 50k images of circle lines, as target 50k images of filled colored circles and corresponding prompts for each target. See Figure 1 for an example.

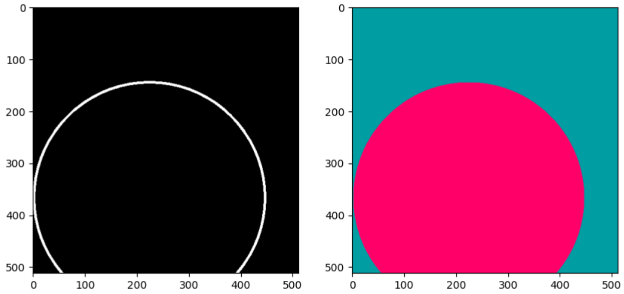

Figure 1. ”hot pink circle with dark turquoise background”

Type	My Laptop	ColabPro	GCP
System RAM	16	25.5	56
GPU RAM	8	16	2*16
Disk	775	166	200

Table 1. Computational Capacity in GB

3.2. Pre-Processing

Thankfully, the fill50k dataset is already well-built so most of the pre-processing comes from setting up an adequate environment that can hold in terms of RAM and GPU. See Table 1 for the details of my computational capacities.

4. Experiments

4.1. Control Stable Diffusion to fill a circle with colors

Based on the available dataset mentioned above, I am running a first experiment, as described in the GitHub for ControlNet here: https://github.com/lllyasviel/ControlNet/blob/main/docs/train.md, to add the condition of filling the input circles with colors. In order to do this I am doing the following:

- Clone the ControlNet Github repository into Colab
- Download the stable diffusion model weights and the dataset and load into Google Drive
- Fix the directory structure to properly attach a control net to the stable diffusion model. See Figure 2 for an example.
- Train the model (I’m currently at this step)
- Evaluate the model on test set
- Analyze results and areas for improvements

4.2. Control Stable Diffusion to do something else

Currently, I am trying to make my first experiment run with the available computational power, and then ideally I would like to make something more specific.

5. Results

I’m currently stuck with CUDA out of memory error on Colab Pro and I’ve submitted access to more GPU RAM with GCP, and I’m waiting to get approved.

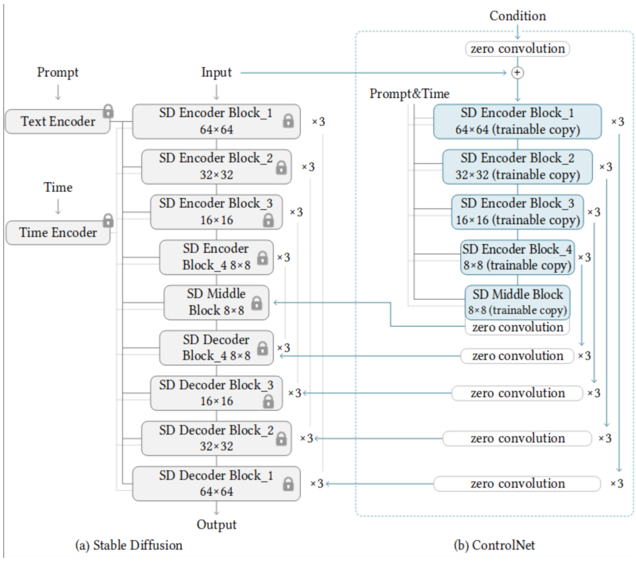

Figure 2. Architecture of control net attached to stable diffusion model

6. Plan for the next weeks

6.1. Computational Power

My goal is to focus the model so that it does not crash my Colab Pro, and leverage the Google Credit that we've been given to train the model.

6.2. Results & Analysis

I'd like to have clear results for the final output and evaluate my results through qualitative evaluation, and if time allows, find quantiative metrics in the literature.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[2] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2