Training ControlNet on Foreground Silhouette and Simple Circle Control Conditions

Annie Chen
MIT
Cambridge, MA
ac134@mit.edu

Abstract

ControlNet has been transformative in the way people use image generation models and has many applications, including data augmentation, data visualization, as well as artistic uses. This paper aims to improve upon the existing flexibility of ControlNet by introducing new conditions to train a ControlNet on. After considering many potential new conditions, I settled on two. I train ControlNet to control Stable Diffusion on the conditional controls, foreground silhouette and circle filling. For data, I utilize existing datasets, performing preprocessing to tailor the data to the project's specific analysis needs. I show that training a ControlNet on these conditions is promising on datasets as small as 1k images. I also highlight some of the challenges with training a large, 350 million parameter ControlNet, with limited computational power and training data.

1. Introduction

Image generation is an integral part of computer vision and has many diverse applications, from improving data augmentation [7] to aiding creative fields [8]. Although text-to-image models provide a great amount of creative freedom and control [1], they are lacking in that (1) users need to be skilled in prompt generating and using specific language and (2) even with good prompts, aspects like spatial composition of the image can't be controlled.

ControlNet is an end-to-end neural network architecture that improves upon existing text-to-image diffusion models by leveraging their robust encoding layers as a strong foundation to learn a diverse range of conditional controls. It works by freezing the original diffusion model's weights, making a trainable copy of the model's encoding layers, and connecting them together with zero convolution layers, with weights initialized to zeros [5]. Then, the whole model is trained on various conditional controls in the form of an image paired with text prompts.

Through this project, I aim to further improve upon the flexibility of ControlNet by training a ControlNet which has been applied to Stable Diffusion V1.5, on a new conditional control, foreground silhouette, as well as a simpler control, circles. I aim to give artists or users of image generation models greater mastery over their creations. With the foreground silhouette condition, the user can simply input the overall shape of their image foreground, with no specifications for the background. This allows for a simpler control image, where the user doesn't need to create a well-done sketch as input.

2. Related Work

2.1. Text-to Image Translation

Generative adversarial networks. Historically, generative adversarial networks (GAN's) have been the main model used to tackle the task of image generation from text [10]. GAN's provide a less data-heavy way to learn deep representations. They work by training two neural networks, a generator and a discriminator, simultaneously, where the generator creates new images while the discriminator tries to distinguish if the image is an existing or generated image. This continues until the discriminator can no longer distinguish between generated and real images. GAN's can produce images quickly, but may fall into the trap of mode collapse, or reproducing the same images which have fooled the GAN's discriminator.

Diffusion Models. More recently, diffusion models have gained popularity due to their ability to generate extremely realistic images. Diffusion models are latent variable models, which work to map observed data to a lower dimensioned latent space to capture the underlying structure of the data. Noise is gradually added to an image until it's indistinguishable from pure noise, and the model is trained to recover the original image from the noise [1].

Diffusion models often surpass GAN's in producing realistic, unique images, and don't fall prey to mode collapse [1]. However, they take more compute power and time to

train.

2.2. Existing ControlNet Conditional Controls

ControlNet has been successfully tested with various conditioning controls, e.g. scribbles, fake scribbles, depth maps, segmentation maps, human pose, normal maps, anime line drawings, HED maps, and combinations of them as well [3].

ControlNet, applied to Stable Diffusion specifically, has been proven to robustly interpret semantics with various prompts and images. Under various conditions, such as with no prompt, insufficient prompts, conflicting prompts, and perfect prompts, ControlNet succeeds in generating accurate images [3].

3. Method

I train the ControlNet model on two conditions, foreground silhouette and circle filling, to investigate whether they will yield promising results in aiding text-to-image generation. Foreground silhouette is a new condition that I decided on after considering many other new conditions. The circle filling condition is one recommended by the authors of ControlNet [3] to practice training ControlNet on. I first briefly describe the structure of ControlNet applied to Stable Diffusion in section 3.1, then run through the process of data preprocessing in section 3.2. I elaborate on my training process in 3.3, explain the evaluation process in 3.4, and highlight some challenges encountered in 3.5.

3.1. ControlNet Structure

ControlNet works by expanding upon the structure of Stable Diffusion, essentially freezing the weights and layers of Stable Diffusion and making a trainable copy of them. Stable diffusion has an encoder with 12 blocks, a middle block, and a skip-connected decoder with 12 blocks. The trainable copy, shown on the right in Figure 1, has the 12 encoding blocks and the middle block from Stable Diffusion. The trainable copy has all its weights initialized to zero and is connected to the Stable Diffusion model, shown on the left in Figure 1.

3.2. Dataset Preparation

ControlNets are trained on datasets with a source image, a target image, and a prompt. To train ControlNet on two new conditions, I looked into existing publicly available datasets to modify to fit the ControlNet structure. First, for the foreground silhouette condition, I took one thousand silhouette and real image pairs from the dataset CAMO [12] and wrote prompts for each of them, with some help from ChatGPT. I chose to use this dataset since it had many silhouettes to real image pairs. I also converted each image into PNG format, resized them to be 320x320 (as ControlNet only works with images with dimensions that are multiples of 64x64) and renamed them to work with the json file I created. Lastly, I organized the data into the structure ControlNet can read. Figure 2 shows example training data for the foreground silhouette condition and Figure 3 shows example training data for the circle filling condition.

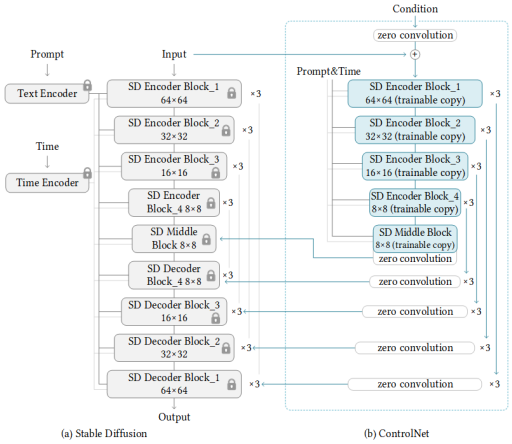

Figure 1. Structure of ControlNet.

3.3. ControlNet Training

To train the ControlNet, two different approaches were used. The first was using a virtual machine in Google Cloud Compute Engine. Due to limited compute resources in every US region, I wasn't able to access Nvidia V100 or A100 GPU's; the most powerful GPU I could use was the Nvidia T4 GPU. Even with Nvidia T4, I frequently had to request quota increases to get enough compute to start training the model. I was able to train the ControlNet to 25 Epochs with 1000 source and target images and prompts, which took 14 hours. I looked into Amazon AWS cloud briefly, but ran into similar compute problems.

When training, to account for the limited computation resources, I trained the model on 1k data instances. Pytorch lightning, which is used to run the training, has a default of 1000 epochs, but I changed it to 25 epochs for manageability. Still, each epoch took upwards of half an hour, so 25 epochs took more than 13 hours.

Since training in Google Cloud wasn't as fast as I preferred, I also tried the Google Colab approach and signed up for Colab Pro, hoping to be able access A100 GPUs. I moved the 13 GB size file containing the ControlNet model and data model, mounted google drive and connected to Google Python 3 Google Compute Engine Backend with A100 GPU. Then, I set up miniConda and created a training environment as defined by the ControlNet environment.yaml (with Pytorch 1.12, CUDA 11.3 and other required modules). In Google Colab, each epoch took around 6 minutes, and I was able to train the model with 65 epochs.

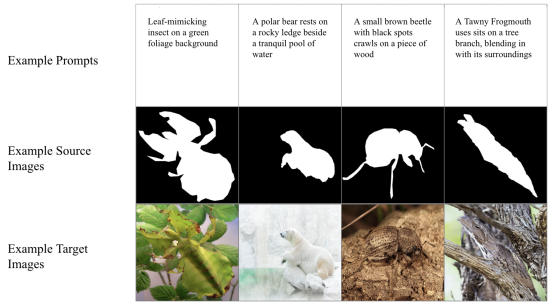

Figure 2. Example training data for the foreground silhouette conditional control.

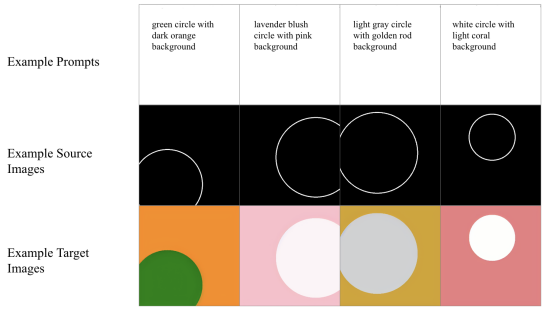

Figure 3. Example training data for the circle filling conditional control.

In both cases, to reduce RAM usage and to speed up run time, I made a few modifications to the original ControlNet code. First, I set the batch size to 1 to reduce memory usage. I also set save_memory and only_mid_control to True as well. I also installed xformers, a module to increase speed and reduce memory usage, but ended up not using it due to incompatibilities with module's pytorch version.

3.4. Performance Evaluation

For both of the conditional controls, I train the ControlNet for 75 epochs, with the modifications to reduce RAM usage and speed up run time stated in Section 3.3 and perform a qualitative analysis of the results. More details are in Section 4.1.

3.5. Challenges

Throughout this project, I ran into quite a few challenges, mostly due to limited computation power and readily available training data. Training on new conditions means that datasets aren't readily available to train on. Additionally, foreground silhouettes differ from many other conditional controls in that silhouettes can't be directly derived from the original image with image transformations, in contrast to controls like edge maps, depth maps, HED maps, etc.

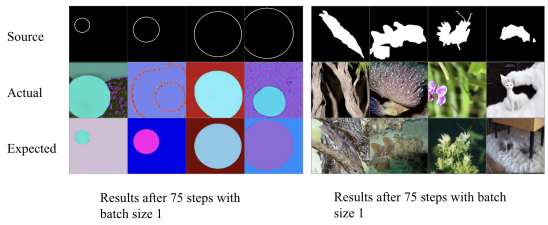

Figure 4. Results after 75 epochs of training with a batch size of 1 and training dataset size of 1k images pairs.

While preprocessing data, I had to manually label 1000 images for my dataset, which was both time consuming and quite laborious. I had tried some image-to-text models to write prompts, but these ended up being unsuccessful. Additionally, since the implementation of ControlNet [3] only takes image inputs with dimensions that are multiples of 64x64, I also spent a while resizing my images.

During training, the lack of access for more reliable and capable GPUs (A100 was recommended for training ControlNet [3]) made training the ControlNet difficult and time consuming. I experienced frequent timeouts and disconnections from Google Colab, due to lack of activity, which was an issue for training the model overnight. I also ran into incompatibility issues with my environment, but that's a standard issue due to various library updates and environment differences.

4. Experiment

We use an implementation of ControlNets with Stable Diffusion to test the new conditions, foreground silhouette and circle filling.

4.1. Qualitative Results

Figure 4 shows the generated images with both the silhouette and the circle filling prompt settings after 75 epochs of training with a batch size of 1 and training dataset size of 1k images.

4.2. Limitations and Future Research

As stated in Section 3.5, the greatest limitations for this project was limited computational power. Due to this, I wasn't able to fully train my models, and I wasn't able to train them on a large enough dataset to yield ideal results. With more training and more data in the future, I believe that the foreground silhouette control will yield promising results. Figure 5 shows a comparison of my training results with the training results of the ControlNet authors [3]. It shows that with more training, as well as a larger training dataset, the model improves greatly.

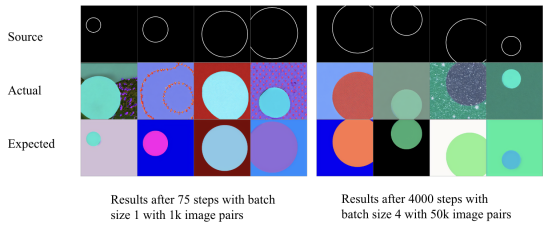

Figure 5. Comparisons of my training results to training results from the ControlNet authors [3].

5. Conclusion

In this study, I trained ControlNet on the conditional controls, foreground silhouette and circle lines. While I had limited findings due to computational power limitations, I also show that foreground silhouette is a promising control condition, which can be validated with more training, larger datasets, and larger batch size. I also highlight some of the challenges with training a large, 350 million parameter ControlNet, with limited computational power and training data. For further research, I also suggest a new condition, color segmentation. Initially when planning my project, I had planned on training ControlNet on a color segmentation conditional control, which would allow users to control which colors go where on their image.

References

[1] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar, Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. 2022.

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In CVPR, 2022.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. arXiv preprint arXiv:2211.09800, 2022.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj¨orn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

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

[6] AlBahar, B., Lu, J., Yang, J., Shu, Z., Shechtman, E., and Huang, J.-B. Pose with Style: Detail-preserving pose-guided image synthesis with conditional stylegan. ACM Transactions on Graphics, 2021.

[7] Tanaka, Fabio Henrique Kiyoiti Dos Santos, and Claus Aranha. Data augmentation using GANs. arXiv preprint arXiv:1904.09135. 2019.

[8] Hyung-Kwon Ko, Gwanmo Park, Hyeon Jeon, Jaemin Jo, Juho Kim, and Jinwook Seo. Large-scale Text-to-Image Generation Models for Visual Artists' Creative Works. In Proceedings of the 28th International Conference on Intelligent User Interfaces (IUI '23). Association for Computing Machinery, New York, NY, USA, 919–933. 2023.

[9] Stability. Stable diffusion v1.5 model card, https://huggingface.co/runwayml/stable-diffusion-v1-5, 2022.

[10] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," in IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65. Jan. 2018.

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj¨orn 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.

[12] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto. Anabranch network for camouflaged object segmentation. CVIU, 184:45–56, 2019.

[13] Divya Saxena and Jiannong Cao. 2021. Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions. ACM Comput. Surv. 54, 3, Article 63 (April 2022), 42 pages. https://doi.org/10.1145/3446374