Reimplementing ControlNet: Adding Conditional Control to Text-to-Image Diffusion Models

Qingyu Wang, Shuyi Qi, Yingtong Yu

Introduction

We reimplemented ControlNet, a method that allows for adding spatial conditioning controls to text-to-image diffusion models, which is a stable diffusion model in our case. We chose this paper because we were interested in stable diffusion models and ControlNet and wanted to explore how they could be enhanced with additional control signals. This is a structured prediction problem in the field of computer vision. The original stable diffusion model only takes in a prompt while ControlNet takes in both a prompt and a hint image to generate images.

Methodology

The original ControlNet paper comes with a toy dataset called fill50k, which contains various shapes like circles of different colors. We trained our ControlNet model on the FaceSynthetics dataset from Microsoft, which contains synthetic face images and their corresponding captions and segmentation. We split the data into a train dataset (80%) and a test dataset (20%). Because of performance issues, we used 800 of the dataset for training.

We built our TensorFlow version of ControlNet based on the StableDiffusion model from keras_cv. We reused the code from keras_cv.models.StableDiffusion and added ControlNet on top of it. To add a ControlNet to a neural network block, we lock the original block, along with its trainable copy and connect them together using zero convolution layers, i.e., 1 × 1 convolution with both weight and bias initialized to zero.

Results

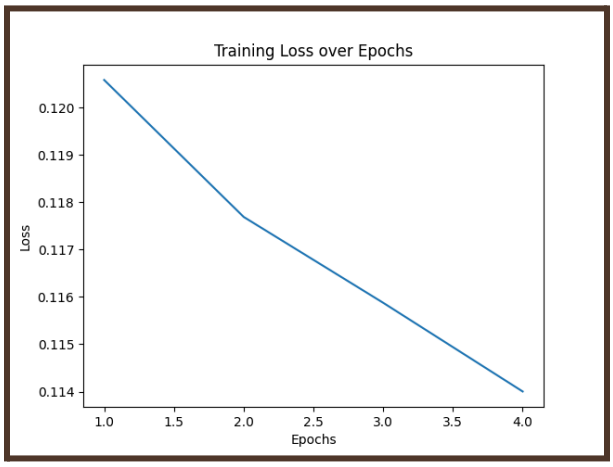

Figure 1: Training Losses over 4 Epochs

From Figure 1, we can see that we have achieved decreasing training losses over 4 epochs, with a low enough value in the first epoch. It exhibits a trend for decreasing more as we potentially train for more epochs, though we did not have enough computational resources to train for this project at this stage.

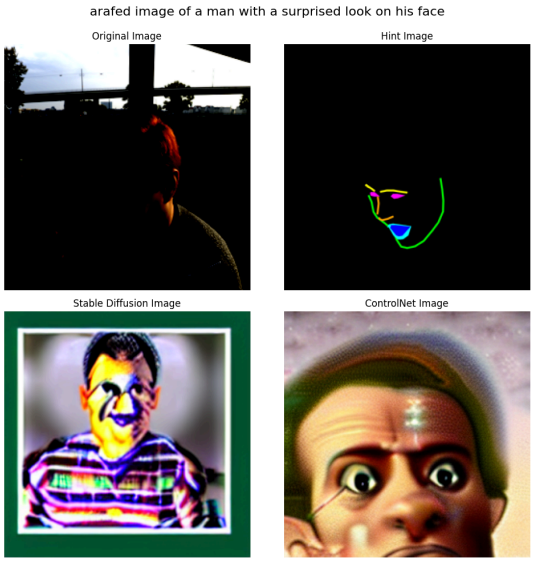

Figure 2: Comparison of the Original Image, Hint Image, Image Generated by Stable Diffusion and Image Generated by ControlNet

From Figure 2, we can see the image generated by the stable diffusion model from the keras_cv package and the image generated by our ControlNet model, along with the original image and the hint. We obtained a CLIP score of 0.27959815 for stable diffusion and 0.28030834 for ControlNet. In addition, we obtained a FID score of 473.41363421419203 for stable diffusion and 402.6039444818882 for ControlNet. A higher CLIP score and a lower FID score means better results so our ControlNet has improved the keras_cv.models.stable_diffusion model.

Challenges

Our major challenge was to reimplement the codebase of ControlNet in TensorFlow, while the codebase that came with the paper was in PyTorch. Most of our time was spent on implementing the TensorFlow version from scratch.

Our second challenge was to write data processing scripts and build a pipeline for training and testing. For example, because stable diffusion itself is already a large model, it is in our best interest to save the model weights after a certain number of epochs and load the weights to resume training. Our pipeline also handles training, inference and testing including calculating the FID scores and CLIP scores of stable diffusion and ControlNet.

Our third challenge was to obtain enough computational resources for training and running the model. It took us around 7 hours to complete training for 4 epochs on the FaceSynthetics dataset so we reduced the size of the dataset that we used for training. We tried to add jit and

mixed precision to our model but jit did not seem to work with Oscar so we did not end up using it.

Reflection

Overall, our project turned out to be satisfactory because we achieved our target goal, which is to make our reimplementation work on the FaceSynthetics dataset, though we trained on 800 of the FaceSynthetics dataset due to limitations in the computational resources and more inference experiments still need to be run. Our stretch goal was to make the CLIP and FID scores match what was included in the original paper. In the original paper, they achieved the same CLIP scores between stable diffusion and ControlNet while the FID score of ControlNet is higher than that of stable diffusion. Our model also achieved very similar CLIP scores for stable diffusion and ControlNet (0.27959815 and 0.28030834). And we achieved a lower FID score for ControlNet. Overall, we would say that we partially achieved our stretch goal.

The model worked out the way we expected it to because from Figure 2 we can see the image generated by Controlnet has an obvious surprising look on the man's face and this image contains traits from the hint image while achieving a relatively good CLIP score.

Our initial plan was to train on the entire FaceSynthetics dataset but with limited computational resources, a large stable diffusion model and the seemingly incompatibility of jit with Oscar, we decided to reduce the train dataset size to 800. If we had more time, we would have investigated more optimization strategies to train the model on the full dataset and run more inferences.

For one of us, this was their first project in computer vision and we all learned about how both stable diffusion and ControlNet work. We also learned that obtaining computational resources is a big part of deep learning so looking into ways to optimize the model would be beneficial. Besides implementing the model itself, we realized that there are also many other aspects of the training and testing pipeline to implement.