Overview

For Assignment 3, we decided to experiment with the ControlNet model from the paper "Adding conditional control to text-to-image diffusion models" by Zhang and Agrawala [5]. ControlNet is able to generate images from text prompts using pretrained diffusion models by adding additional weights that guide the outcome. To achieve this, they provide pretrained ControlNet models that each work for a specific type of conditional input. This includes user generated scribbles or preprocessing of existing images such as depth estimation, semantic segmentation or line detection.

We experimented with the different models and their ability to guide image generation. For the tests we describe in this report, we focused on testing whether we can use ControlNet for synthetic data generation, using the example of tennis court recognition.

Installation & Setup

The code for ControlNet was downloaded from the official python implementation[1] and placed into src/ControlNet.

The pretrained models were downloaded from HuggingFace[2] and put into the src/ControlNet/models folder. As the pretrained models are quite large, we did not include them in our submission.

A python virtual environment can be created using the requirements.txt file we added, with updated pytorch versions compared to the original implementation. For faster inference and even lower memory consumption,we also installed the xformers library (see Github issue [4]).

Changes

To allow inference with 8GB VRAM graphics cards, we use the low memory mode, setting the save_memory variable under config.py to True.

To get the code running with the newer package versions, the imports

```
from pytorch_lightning.utilities.distributed import rank_zero_only
```

had to be changed to

```
from pytorch_lightning.utilities.rank_zero import rank_zero_only
```

in two seperate files:

- src/ControlNet/cldm/logger.py
- src/ControlNet/ldm/models/diffusion/ddpm.py

Usage

To run our experiments, we used the provided gradio Scripts(gradio_*.py).

Tests

Synthetic data generation

We want to see whether we can use ControlNet to create synthetic datasets for keypoint detection models.

The idea for this test came from previous experiments with standard diffusion models. We encountered two main problems when generating such data using diffusion models without any conditional input:

- The created images are not correct
- We not have labels for the generated data

Figure 1: Tennis court generated using stable diffusion 1.5

Our hypothesis is that ControlNet can overcome those problems, as we have conditional input from real images. This should allow ControlNet to avoid generating wrong geometry. Furthermore, if we know the labels (for example the pixel coordinates of the desired keypoints) for the real images, we can use those labels for the generated output. Sometimes, the preprocessing automatically yields the desired labels, for example, when using estimated poses from the original image as the conditional input.

We will test this hypothesis using the example of tennis court recognition. Note that we will not perform quantitative tests or train actual models using the generated data.

All the conditional inputs for this task are based on the same reference image, which was taken from [3].

Figure 2: Tennis court reference image

Line Annotations

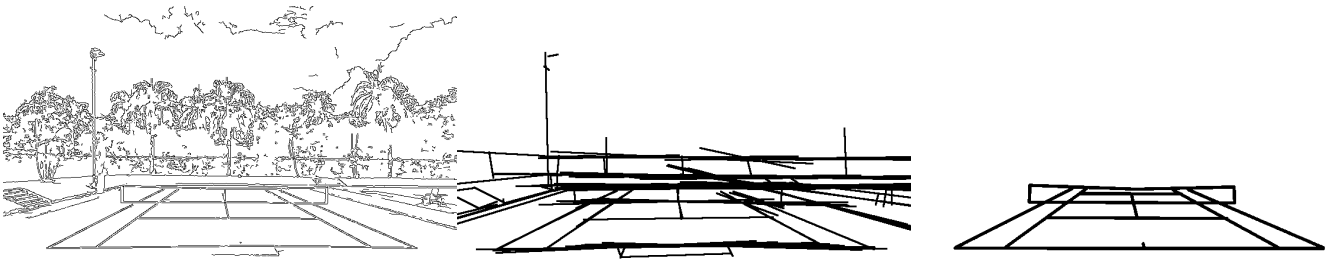

Figure 3: Different conditional inputs. Left: Canny edge detection, Middle: Hough lines, Right: Scribble

We obtained the best results when using conditional inputs that preserve much of the tennis court information.

Figure 4: Generated image using Canny edges as conditional input

When using the Canny edges as conditional input, we see that the generated input closely matches the reference image in the background due to the high number of edges in the conditional input image. This is not the case when using the scribble as conditional input, as the scribble only contains information about the tennis court itself.

The following images were all generated using the following prompt:

```
indoor tennis court, no players, concrete walls, ceiling lighting
```

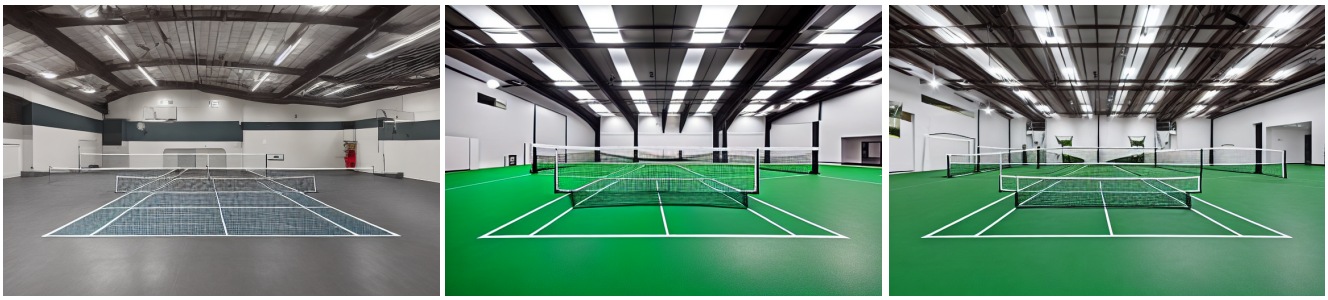

Figure 5: Images generated using the scribble as annotation, with different parameters for guidance and number of diffusion steps

We did not notice siginificant differences in the images when altering the guidance scale or number of diffusion steps.

In the above Images as well as in other generated images, we noticed that the biggest problem for the model is the accurate generation of the net.

Figure 6: Wrong court geometry when using Hough lines as conditional input

Stock photo text prompt

We tried generating realistic images by adding "stock photo" to the text prompt.

Using the following prompt: `red clay tennis court, stock photo, no players` we noticed that this can generate artifacts, as seen in Figure 6.

Figure 7: Generated image with visible artifact due to "stock photo" text prompt

Semantic Segmentation Annotations

Using semantic annotations as the conditional input for the diffusion model did not yield good results for this task. This makes sense, as the semantic segmentation does not contain any information about the geometry of the tennis court but only where the tennis court is compared to the background.

Figure 8: Left: Semantic segmentation annotation Right: Generated image

Conclusion

While we see that the generated images from ControlNet are much more accurate compared to images generated with unconditioned diffusion models, the geometry is still not perfect. The lines of the tennis court are usually perfectly placed on top of the annotation lines. Sometimes there are additional lines or even nets that are not supposed to be there.

More in-depth experiments or better knowledge of prompt engineering could perhaps increase the quality of the generated images. In conclusion, we can say that ControlNet could be a good tool for generating synthetic data. It is not perfect and might require some experimentation and manual confirmation of the generated images to curate a high quality dataset.

Better results could be obtained by training a specific model with a custom conditional input for this task instead of using pretrained models.

References

[1] https://github.com/lllyasviel/ControlNet, accessed on 19.06.2023

[2] https://huggingface.co/lllyasviel/ControlNet, accessed on 19.06.2023

[3] https://freerangestock.com/photos/39555/tennis-court.html, accessed on 19.06.2023

[4] https://github.com/lllyasviel/ControlNet/issues/3, accessed on 19.06.2023

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