Stable Diffusion, Do? Do!

Jang seung woo

Abstract—It can't write in Korean / www.overleaf.com
I wanted to create an image through deep learning, and I tried a model that creates an image that fits it when I give a prompt. As a result, it was confirmed that the image was printed according to Prompt, but it did not work well for areas that were not learned.

1 ABSTRACT

USING the Stable Diffusion model, create an image between two prompts and fine-tune to Dreambooth Import Pre-Trained weight to create an image by building a model pipeline

2 INTRODUCTION

Diffusion models include the diffusion model of deep mind (DVIM), and the "Noise-Conditioning Attention Network (NCAE)", which combines diffusion models with GANs to achieve higher quality image generation. However, we will use the Stable Diffusion model model that can be easily used in keras to see the results. The Stable Diffusion model is a diffusion model that uses a method of slowly diffusing pixels in an image to effectively handle noise or other deformations. Simply put, initially a random noise or transformation image is used as the starting point, and it is repeatedly spread in a controlled manner to achieve the desired result. The downside is that the configuration is difficult and it takes quite a while to create. So I will lower the number of training and proceed with sampling.

3 METHOD

I checked using Stable Diffusion among the diffusion models. Create gif, image by entering 2 Prompts using keras cv.models.StableDiffusion. Fine-Tuning with dreambooth of diffusers git. Create an image using StableDiffusion's Pipeline

3.1 StableDiffusion

How does the experiment work

1) Using StableDiffusion in the keras cv library
2) Dimension after converting input text to embeddings
3) Create intermediate embeddings by linearly interpolating between
4) the first and second embeddings
5) Leverage interpolated embeddings and noise to create images

- prompt1 = "clouds like cotton candy in the sky"
- prompt2 = "clouds like tomato in the sky"
- interpolationsteps = 5

3.2 dreambooth

How does the experiment work

1) Clone the content related to the diffusers github of the hugging face
2) Set up your environment by referring to requirements
3) Check the image to learn
4) Model training after entering factors related to accelerate launch
5) Configure the entire pipeline using Diffusion Pipeline
6) Creating an Image

- prompt = "A photo of sks dog chasing a car"

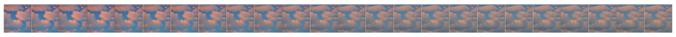

Fig. 1: Enter Caption

3.3 StableDiffusion's Pipeline

How does the experiment work

1) Building a Pipeline with Stable Diffusion Pipeline
2) Use load lora weights to apply weight to other users' pre-learning models
3) prompt, create an image with reference to negative prompt

- prompt="puppy Papillon and Pomeranian"

4 Result

4.1 StableDiffusion

Clouds like cotton candy came out well, but clouds like tomatoes didn't come out, Personally, I expected a cloud in the shape of a tomato, but I could see that only the color would change to a red color. This is presumed to be due to lack of data on tomato-shaped clouds, and it will work well if text is entered as an intuitive object. (Fig 1)

4.2 dreambooth

The data is written with a dog-example chapter 5 image of the huggingface hub. Perhaps because of the lack of training steps due to time, low quality results came out. When creating the model, the learning rate was 2e-6 max train step was 10 and when more was given, a similar image was created in the prompt. If you give enough data images, learning rate, and max train steps, you will get better results. (Fig 2)

4.3 StableDiffusion's Pipeline

Models written with weights from other pre-trained users produced good quality results, but it is assumed that images related only to the prompt in which the model was trained will be produced. I put the dog type separately, but it came out similar, and according to the model's representative image, it came out as a fat puppy. This is speculated that even though the trained model has more than 1GB of capacity, the categories may be small, so we need to bring in various models and write them according to the class. (Fig 3)

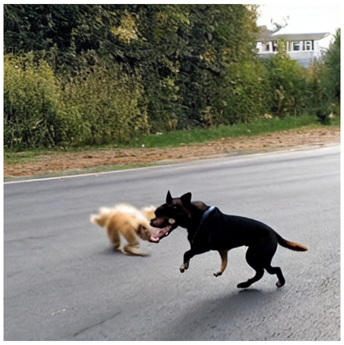

Fig. 2: Enter Caption

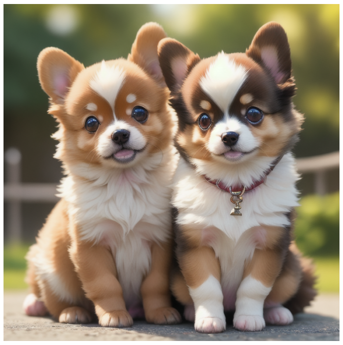

Fig. 3: Enter Caption

5 Acknowledgment

Chat GPT 3.5. Colab. civitai. AIFFEL.