DLCV HW2 Report

B11901040 項達均

October 29, 2024

1

1.1

I implemented the Denoising Diffusion Probabilistic Model using a UNet architecture within PyTorch Lightning. The dataset combines MNIST and SVHN images, and a split training-validation setup with transformations for resizing and normalizing images. The model integrates with Weights & Biases for logging. I used 500 denoising steps and an alpha schedule ranging from 0.999 to 0.99. The conditioning scale is chosen to be 5.

Initially, I used 1000 denoising steps and a smaller alpha schedule, but inferencing was too slow and inaccurate. So I adjusted the parameters in addition to increasing the conditioning scale, which greatly improved the accuracy.

1.2

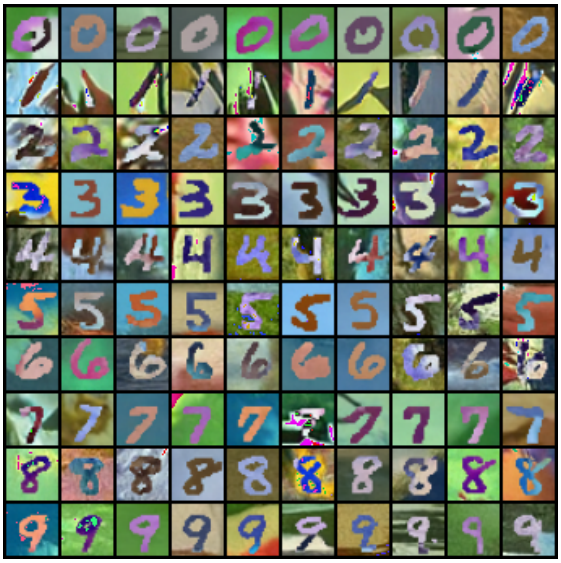

Figure 1: MNISTM Dataset

Figure 2: SVHN Dataset

1.3

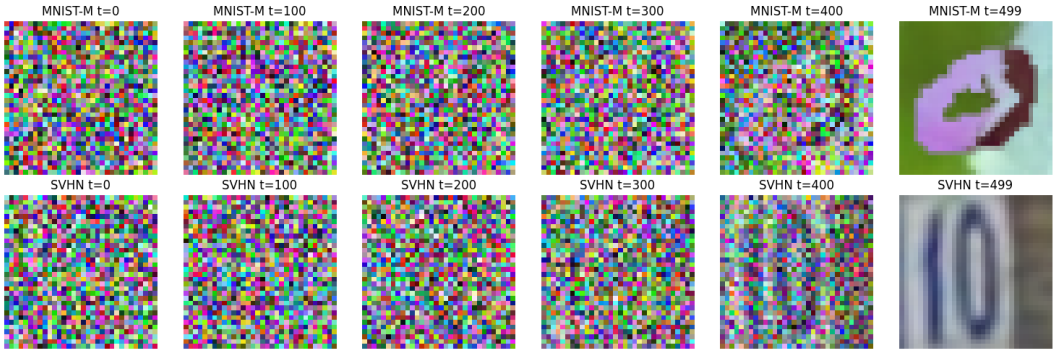

Figure 3: Visualization of Denoising process

2

2.1

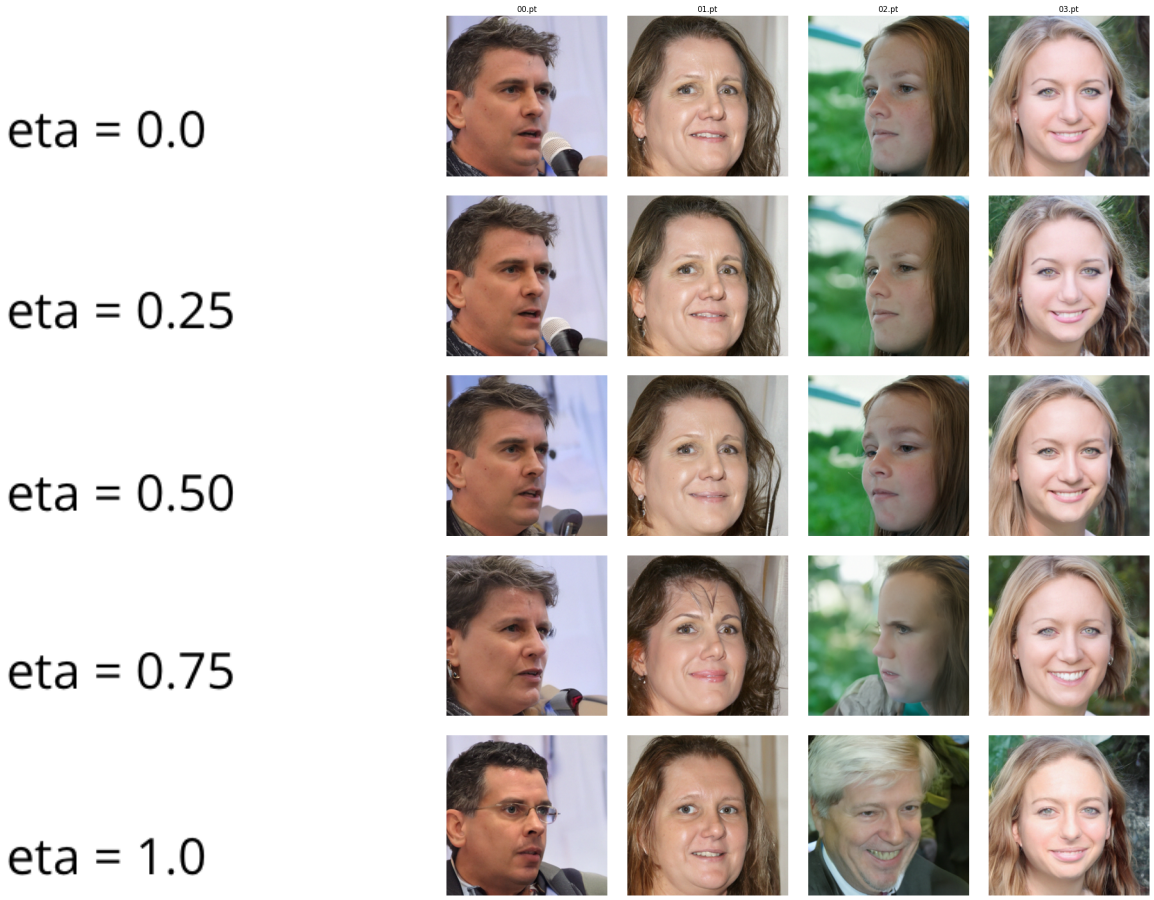

Figure 4: Face images of noise 00.pt 03.pt from different eta

The value of eta controls the stochasticity of the image generation process, the higher the value of eta, the more the image differs from the deterministic one, allowing for more diverse images.

2.2

Figure 5: Spherical Interpolation

Figure 6: Linear Interpolation

With spherical interpolation, the interpolated noise follow the surface of the sphere where the noise vectors lie. As a result, generated face images will transition more smoothly and naturally. However, with linear interpolation, the interpolated noise lies on a straight line between the two noise vectors, not accounting for the spherical space of noise vectors. Therefore, generated images will look more unnatural.

3

3.1

CLIP consists of an image encoder and a text encoder, pretrained using a large dataset of image-caption pairs. During inference, the model computes the cosine similarity between the encoded image and the text embedding. The class with the highest similarity score is chosen as the predicted label.

My experiment resulted in an accuracy of 63.48%. The low resolution of the images makes classification difficult even by the human eye. In the incorrectly labeled cases, the prediction and the ground truth often share similar colors, shapes, or texture.

Figure 7: Correct cases

Figure 8: Incorrect cases

3.2

Figure 9: Prompt: a <new1> next to a <new2>.

When applying textual inversion to multiple concepts, the model is only to generate images of one of the concepts. I read the paper Multi-Concept Customization of Text-to-Image Diffusion, which also suggested that textual inversion struggles with image generation with multiple concepts. In this paper, the datasets corresponding to different concepts are trained jointly. In addition, only a subset of cross-attention layer parameters is fine-tuned, which significantly reduces the fine-tuning time. As their method only updates the key and value projection matrices corresponding to the text features, it can subsequently be merged to allow generation with multiple fine-tuned concepts.