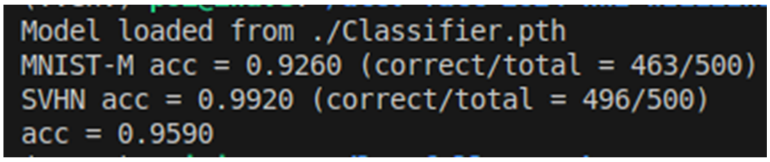
Deep Learning for Computer Vision

NTU 2024 Fall Homework2

電信所碩二 廖珀毅 **R12942009**

- **Problem 1: Conditional Diffusion Models**

    1. **Describe your implementation details and the difficulties you encountered.**

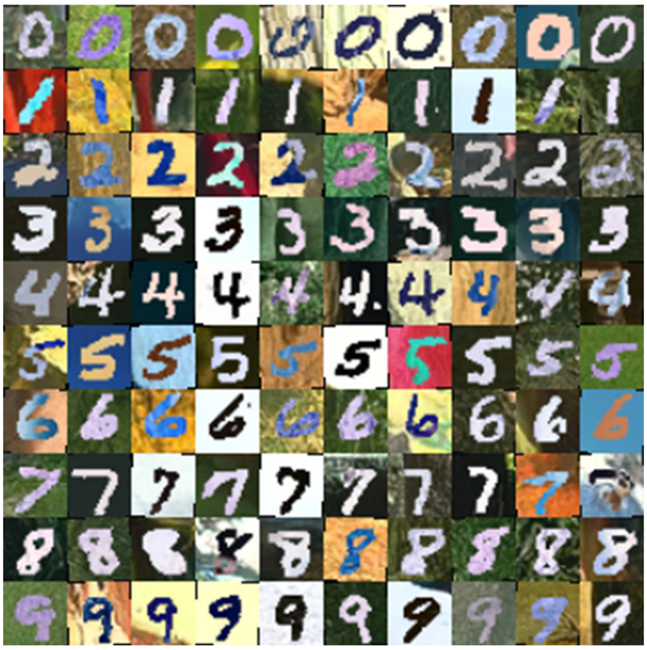


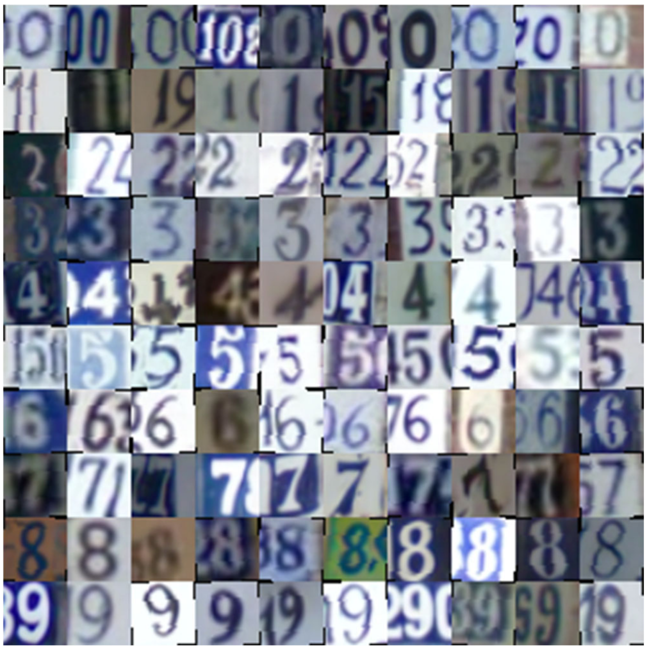
```
Model loaded from ./Classifier.pth
MNIST-M acc = 0.9260 (correct/total = 463/500)
SVHN acc = 0.9920 (correct/total = 496/500)
acc = 0.9590
```

In Problem 1, I modified the UNet from the TAs to implement a conditional diffusion model. This involved adding class labels as input and conditioning the diffusion process on them. During training, I introduced slight rotation augmentation and trained the model to predict noise using MSE loss.

Initially, my custom UNet achieved only 75% accuracy on MNIST-M. After switching to the TA-provided model, MNIST-M accuracy improved to 88%, and SVHN increased from 96% to 99%. I also experimented with the Classifier-Free Guidance (CFG) scale, which controls the trade-off between conditional and unconditional generations. Increasing the CFG scale from 1 to 3 noticeably improved the quality of the sampled images. This increase in CFG scale essentially gives more weight to the class-conditioned generations, making the samples more consistent with the desired labels. As a result, the model performed better in terms of both classification accuracy and the visual quality of the generated images.

In terms of architecture, while I didn't observe a drastic difference in performance between my custom-designed UNet and the slightly modified conditional UNet provided by the TAs, the process of trial and error helped me better understand how the diffusion model operates. More importantly, it highlighted the importance of tuning hyperparameters like the CFG scale.



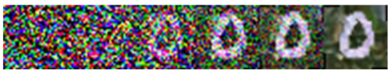


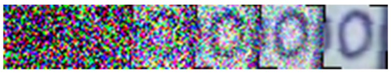
2. **Please show 10 generated images for each digit (0-9) from both MNIST-M & SVHN dataset in your report.**

Both images are provided in the previous pages.

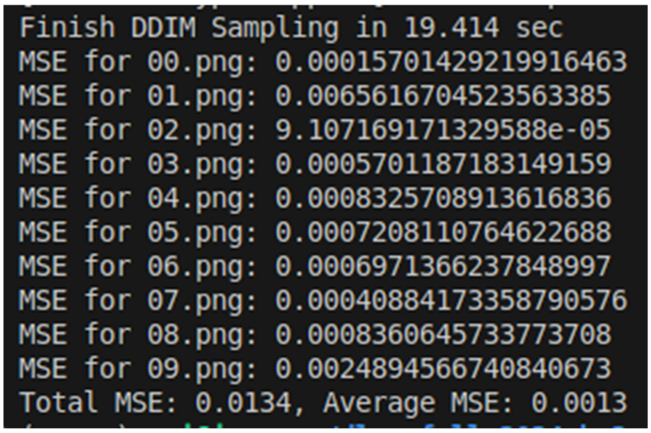
3. **Visualize a total of six images from both MNIST-M & SVHN datasets in the reverse process of the first “0” inyour outputs in (2) and with different time steps.**

The selected timesteps for visualization in this case are 1000, 500, 200, 100, 50, and 0. These intervals provide a clearer view of how the diffusion model progressively denoises the image. Compared to a linear sequence of timesteps, this selection is more effective in showcasing the crucial stages of the denoising process. The smaller intervals toward the end highlight the final, more refined denoising steps. This approach makes it easier to visualize how the model transitions from heavy noise to a clear image.





- **Problem 2: DDIM**



```
Finish DDIM Sampling in 19.414 sec
MSE for 00.png: 0.00015701429219916463
MSE for 01.png: 0.006561670452356385
MSE for 02.png: 9.107169171329588e-05
MSE for 03.png: 0.0005701187183149159
MSE for 04.png: 0.0008325708913616836
MSE for 05.png: 0.0007208110764622688
MSE for 06.png: 0.0006971366237848997
MSE for 07.png: 0.00040884173358790576
MSE for 08.png: 0.0008360645733773708
MSE for 09.png: 0.0024894566740840673
Total MSE: 0.0134, Average MSE: 0.0013
```

1. **Please generate face images of noise 00.pt ~ 03.pt with different eta in one grid. Report and explain your observation in this experiment.**

When Eta = 0.0, which is known as Deterministic Sampling, no additional noise is added at each timestep. The generation process strictly follows the denoising path predicted by the model without any randomness. As a result, the generated images are typically very similar to the ground truth. However, the diversity in the generated images may be limited.

As Eta increases, the sampling process becomes stochastic, where a portion of random noise is added, controlled by the Eta parameter. Higher Eta values result in more stochastic sampling. Among these Eta values, which directly affect the sigma in each step, when Eta is lower than 0.5, the generated images tend to be clearer and more consistent. In contrast, with Eta values larger than 0.5, the images display more diversity and variation.

2. **Please generate the face images of the interpolation of noise 00.pt ~ 01.pt. The interpolation formula is spherical linear interpolation, which is also known as slerp.**

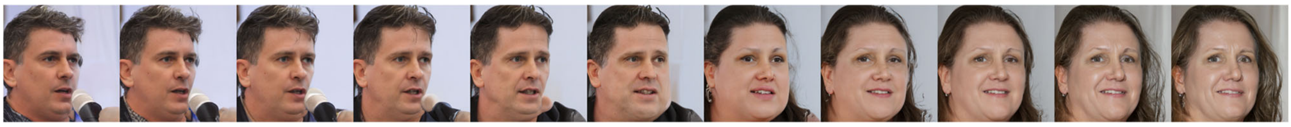
$$\boldsymbol{x}_T^{(\alpha)} = \frac{\sin((1-\alpha)\theta)}{\sin(\theta)}\boldsymbol{x}_T^{(0)} + \frac{\sin(\alpha\theta)}{\sin(\theta)}\boldsymbol{x}_T^{(1)}$$

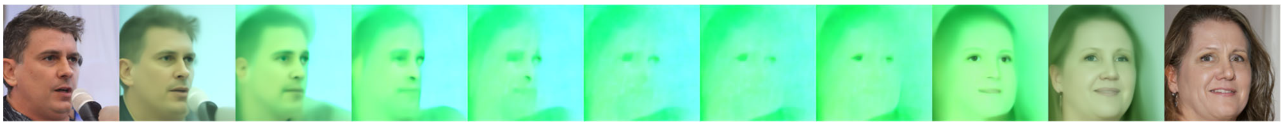
where $\theta = \arccos\left(\frac{(\boldsymbol{x}_T^{(0)})^\top \boldsymbol{x}_T^{(1)}}{\|\boldsymbol{x}_T^{(0)}\|\|\boldsymbol{x}_T^{(1)}\|}\right)$. These values are used to produce DDIM samples.

in this case, α = {0.0, 0.1, 0.2, …, 1.0}.

What will happen if we simply use linear interpolation? Explain and report your observation.

From the following figures below, the top one image applying the slerp interpolation, while the one underneath using linear. From basically observation through the images, slerp interpolation maintains smooth transitions between the faces and keeps the features of the generated images consistent, in contrast, linear interpolation results in artifacts, and the transition becomes less coherent in intermediate stages. It generates unrealistic color changes, as seen in the greenish hue, making the images more distorted. The difference between slerp and lerp is that slerp interpolates the points along the surface of a hypersphere with sin((1−α)θ) and sin(αθ). Considering the geometry of the data in high-dimensional space by taking the angle between the vectors into account.

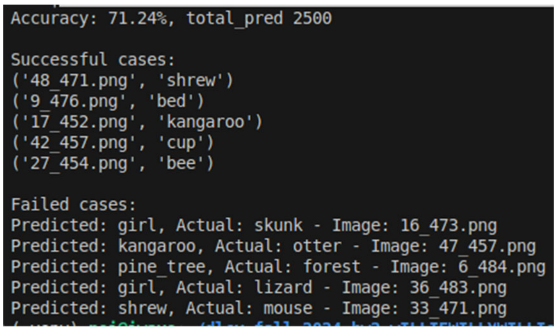




- **Problem 3: Personalization**

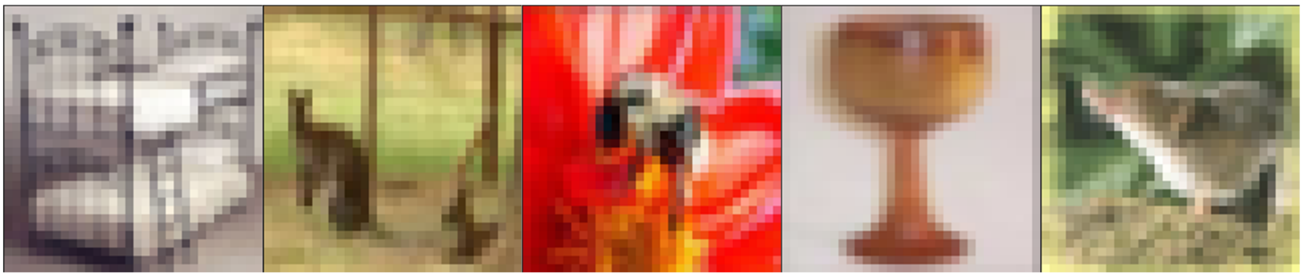
  1. **Conduct the CLIP-based zero shot classification on the hw2_data/clip_zeroshot/val, explain how CLIP do this, report the accuracy and 5 successful/failed cases.**

The CLIP model performs reasonably well in the zero-shot classification task, achieving an accuracy of 71.24%. However, the model's errors highlight its difficulty in distinguishing between visually similar classes, which is a common challenge in zero-shot learning.



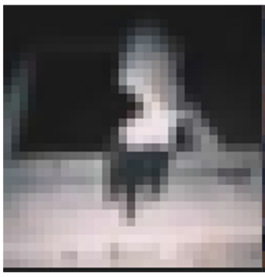
```
Accuracy: 71.24%, total_pred 2500

Successful cases:
('48_471.png', 'shrew')
('9_476.png', 'bed')
('17_452.png', 'kangaroo')
('42_457.png', 'cup')
('27_454.png', 'bee')

Failed cases:
Predicted: girl, Actual: skunk - Image: 16_473.png
Predicted: kangaroo, Actual: otter - Image: 47_457.png
Predicted: pine_tree, Actual: forest - Image: 6_484.png
Predicted: girl, Actual: lizard - Image: 36_483.png
Predicted: shrew, Actual: mouse - Image: 33_471.png
```

Successful Cases



9_476.png 17_452.png 27_454.png 42_457.png 48_471.png

Fail Cases


16_473.png

47_457.png

6_484.png

36_483.png

33_471.png

Misclassifications often occur in cases where the images are visually similar to another class. This points to a limitation in distinguishing fine-grained details in the zero-shot setting, particularly for classes with overlapping or visually similar features.

The image is passed through the CLIP image encoder, producing an embedding representing its visual features. While the text prompts (e.g., "A photo of a [class]") are tokenized and passed through the CLIP text encoder to generate embeddings for each class label. Cosine similarity is calculated between the image embeddings and the text embeddings to estimate how well the image matches each prompt, selecting the closest match.

2. **What will happen if you simply generate an image containing multiple concepts (e.g., a <new1> next to a <new2>)? You can use your own objects or the provided cat images in the dataset. Share your findings and survey a related paper that works on multiple concepts personalization, and share their method.**

In this task ,I use my my as the concept to train the image After loading the checkpoint

```
INFO:__main__:Token ID 49409 updated.

INFO:__main__:Checkpoint loaded from ./epoch600new_ckpt.pth

INFO:__main__:Token ID 49410 updated.

INFO:__main__:Checkpoint loaded from ./epoch0_3_ckpt.pth
```

As we can see after the token from the two checkpoint are being update,no matter the prompt is A photo of <new3>.or <new3> in the style of <new2>. Where the figure below the leftmost is one of the concept, and the middle one is A photo of <new3>, and the rightmost one is <new3> in the style of <new2>.

In my observation through the work, generating images with multiple concepts, which we known as a new token, may lead to visual fidelity and semantic consistency. Recent research, particularly the paper titled "Concept Conductor: Orchestrating Multiple Personalized Concepts in Text-to-Image Synthesis," introduces innovative methods to address these issues.

It apply multipath sampling, which isolates denoising processes to prevent attribute leakage between concepts.nsuring that the characteristics of one concept do not interfere with another. Each single-concept model generates its corresponding subject separately before integration into a composite image. Furthermore, the concept injection technique employs shape-aware masks to define specific areas for each concept's generation. This method allows for the injection of visual features (structure and appearance) into the final image through feature fusion in the attention layers, ensures that each concept is harmoniously integrated while retaining its unique attributes.

Overall, the method in the paper effectively mitigates common failures associated with multi-concept generation such as attribute leakage and layout confusion.