Deep Learning for Computer Vision HW2

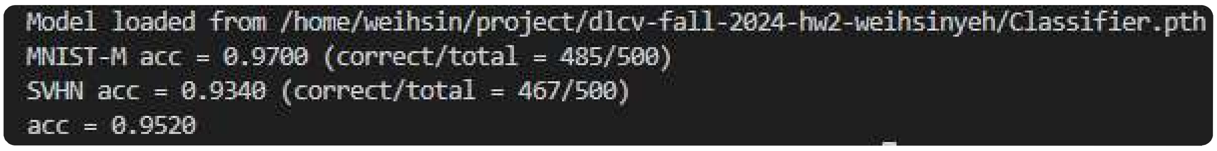
contributer <weihsinyeh (https://github.com/DLCV-Fall-2024/dlcv-fall-2024-hw2-weihsinyeh)>
姓名：葉惟欣
學號：R13922043
系級：資工所 113

作業要求投影片 (https://docs.google.com/presentation/d/1nWH_CmF6iba0kQmi0TV_yI2Emu1EvjrX/edit#slide=id.g28b56d7d974_0_7)

Problem 1: Conditional Diffusion Models (30%) [digit dataset - MNIST-M & SVHN]

evaluation (15%)

epoch	MNIST-M acc	SVHN acc	acc
100	0.9700 (correct/total = 485/500)	0.9340 (correct/total = 467/500)	0.9520



```
Model loaded from /home/weihsin/project/dlcv-fall-2024-hw2-weihsinyeh/Classifier.pth
MNIST-M acc = 0.9700 (correct/total = 485/500)
SVHN acc = 0.9340 (correct/total = 467/500)
acc = 0.9520
```

report

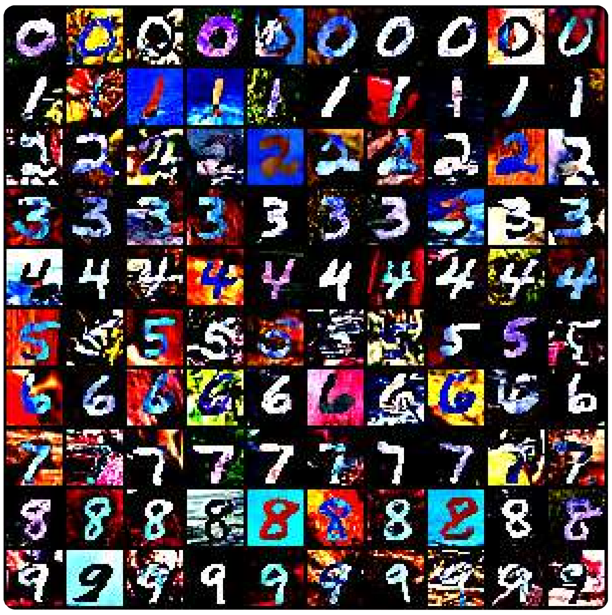
(5%) Describe your implementation details and the difficulties you encountered.

在這個作業中我參考 TeaPearce/Conditional_Diffusion_MNIST (https://github.com/TeaPearce/Conditional_Diffusion_MNIST) 中Unet 的實作，在這個作業我遇到最困難的是要預測出兩個 label ， 一個是datasets 的 label 一個是數字的 label，原先我是使用兩個分類器去實作，一個預測一個datasets，另一個預測數字，但是後來我發現一直訓練不好，所以我就改為 0-9 為預測 MNIST-M 的數字，10-19 為預測 SVHN 的數字，這是因為老師上課提到排列組合的方式，我就想 0 為 MNIST-M， 1 為 SVHN。後面再接一個 0-9 。

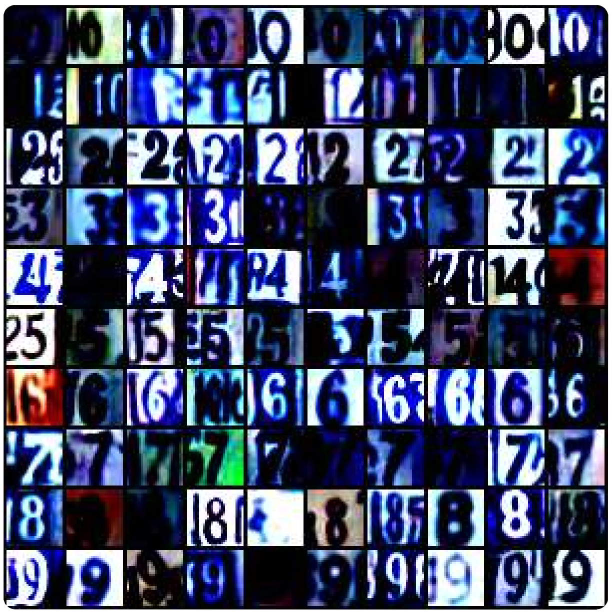
(5%) Please show 10 generated images for each digit (0-9) from both MNIST-M & SVHN dataset in your report.

You can put all 100 outputs in one image with columns indicating different noise inputs and rows indicating
different digits. [see the below MNIST-M example, you should visualize BOTH MNIST-M & SVHN]

MNIST 100 張

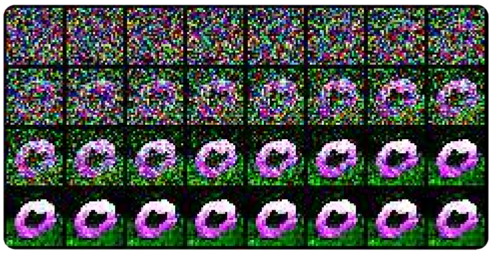


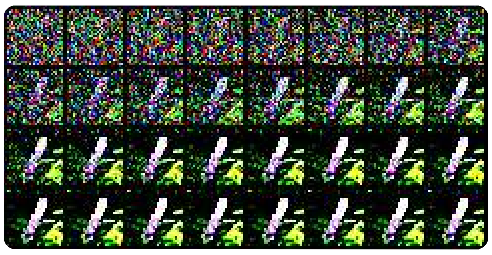
SVHN 100 張

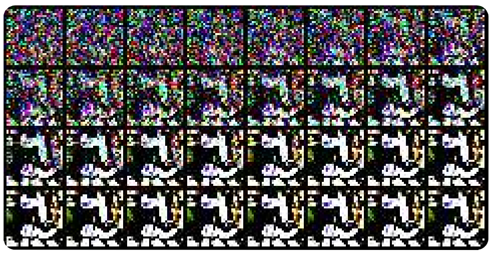


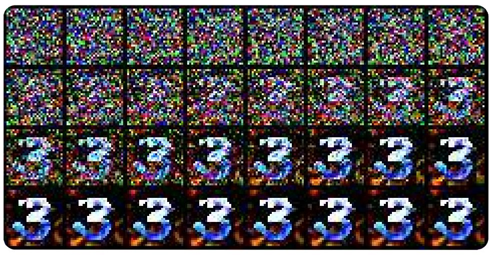
(5%) Visualize a total of six images from both MNIST-M & SVHN datasets in the reverse process of the first "0" in your outputs in (2) and with different time steps. [see the MNIST-M example below, but you need to visualize BOTH MNIST-M & SVHN]

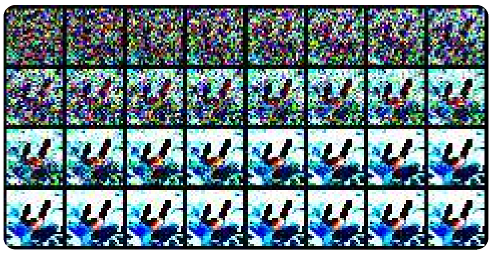
MNIST 0-9

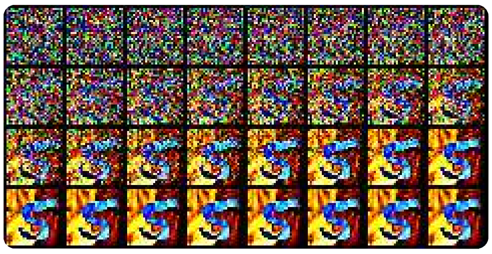


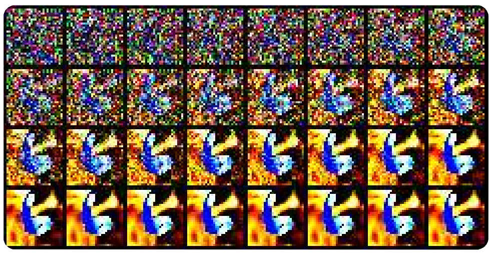


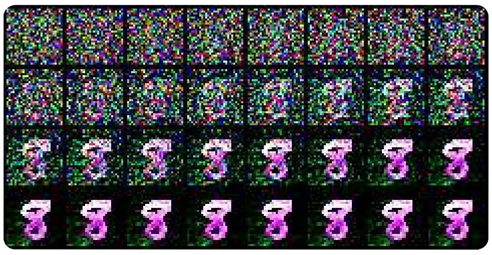


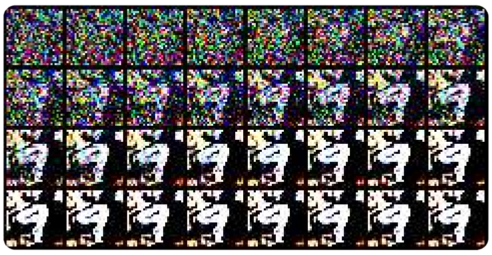




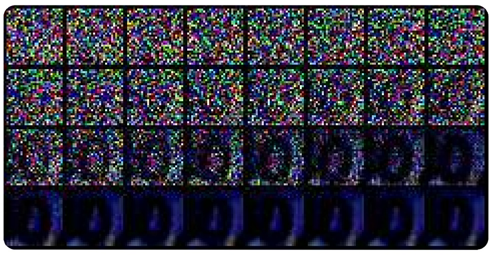


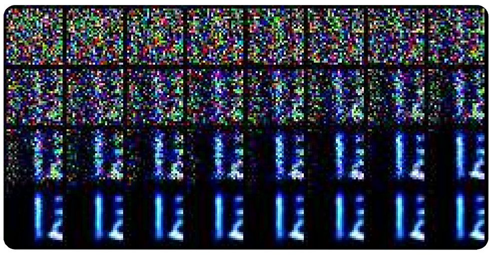


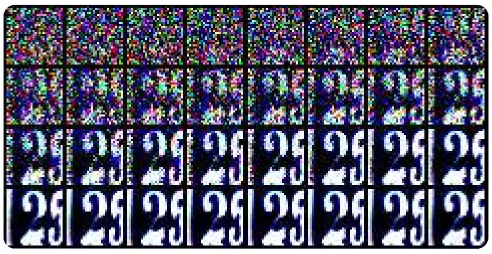


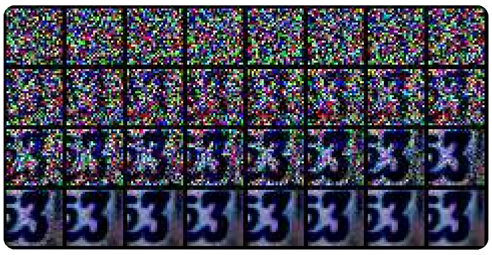


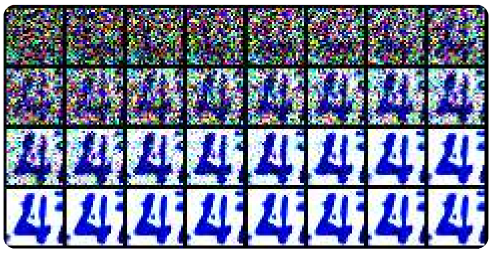
SVHN 0-9

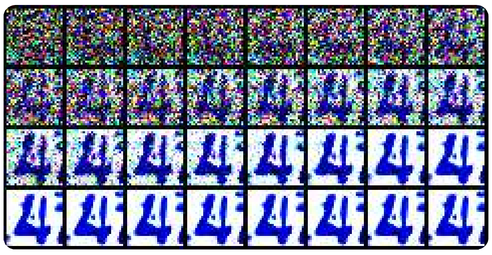


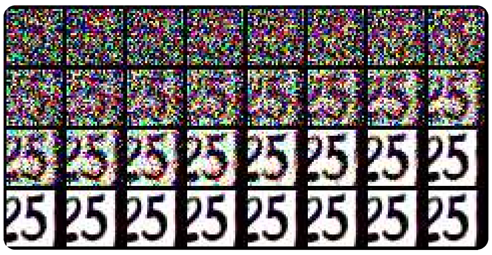


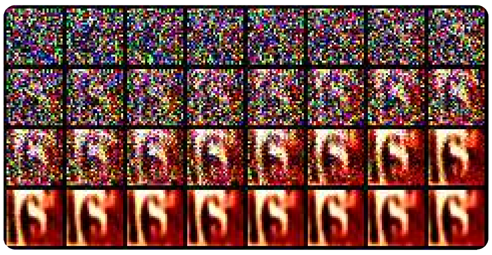


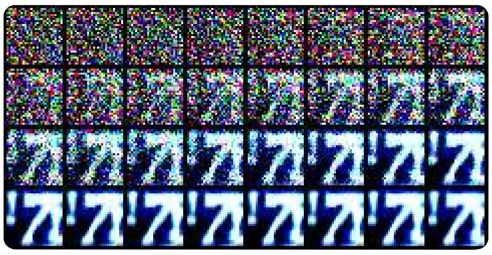


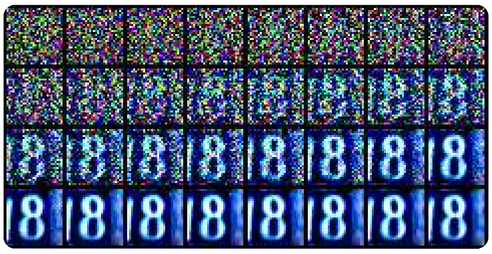


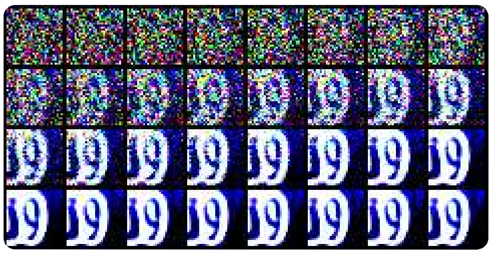












Problem 2: DDIM (35%) [face dataset]

Evaluation (20%)

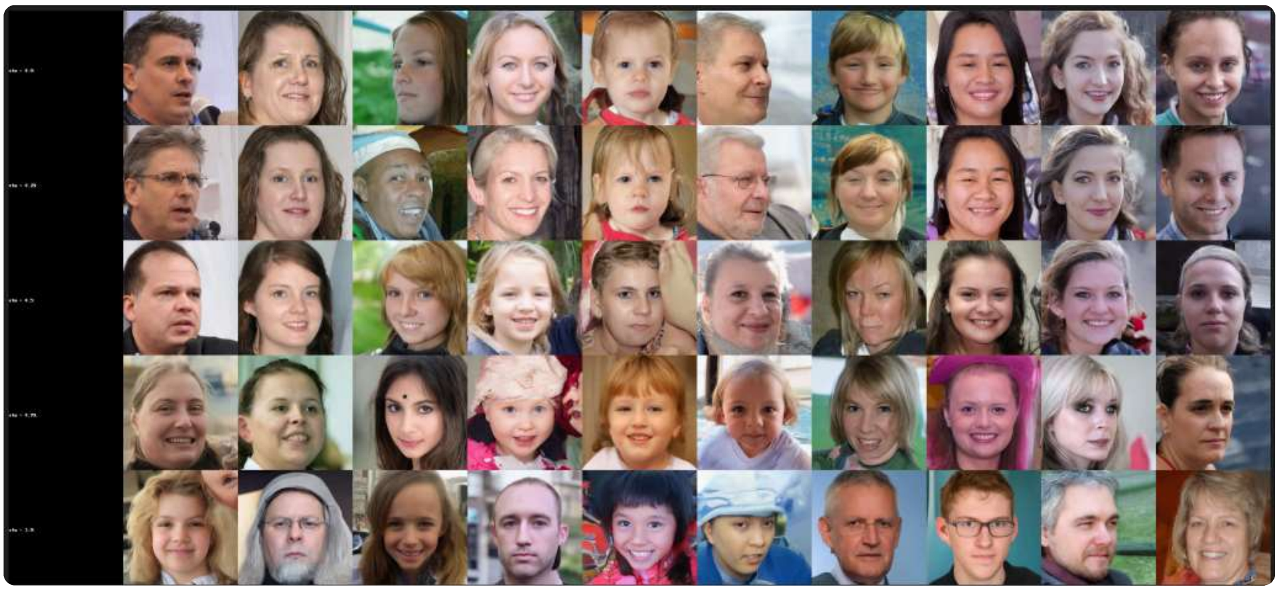
```
MSE for image pair 0: 0.040878
MSE for image pair 1: 0.041499
MSE for image pair 2: 0.067683
MSE for image pair 3: 0.061432
MSE for image pair 4: 0.081599
MSE for image pair 5: 0.047546
MSE for image pair 6: 0.112366
MSE for image pair 7: 0.062302
MSE for image pair 8: 0.048223
MSE for image pair 9: 0.053574
```

(7.5%) Please generate face images of noise 00.pt (http://00.pt) ~ 03.pt (http://03.pt) with different eta in one grid. Report and explain your observation in this experiment. (This following image is just for illustration.)

實驗執行

```
cd PB2
python3 compare.py
```

實驗結果：由上到下是 **0, 0.25, 0.5, 0.75, 1**



(7.5%) Please generate the face images of the interpolation of noise 00.pt (http://00.pt) ~ 01.pt (http://01.pt). The interpolation formula is spherical linear interpolation, which is also known as slerp.

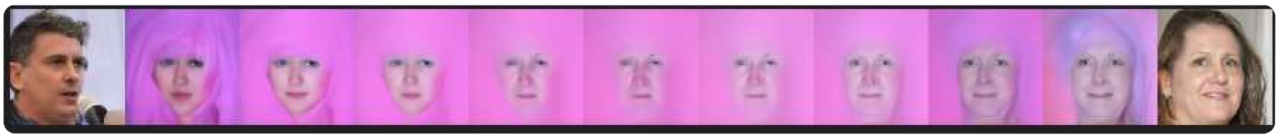
$$\boldsymbol{x}_T^{(\alpha)} = \frac{\sin((1-\alpha)\theta)}{\sin(\theta)}\boldsymbol{x}_T^{(0)} + \frac{\sin(\alpha\theta)}{\sin(\theta)}\boldsymbol{x}_T^{(1)}$$

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.

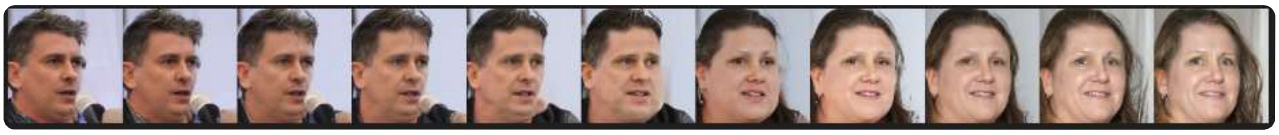
in this case, α = {0.0, 0.1, 0.2, ..., 1.0}.
What will happen if we simply use linear interpolation? Explain and report your observation. (There should be two images in your report, one for spherical linear and the other for linear)

實驗結果

linear 的結果



slerp 的結果

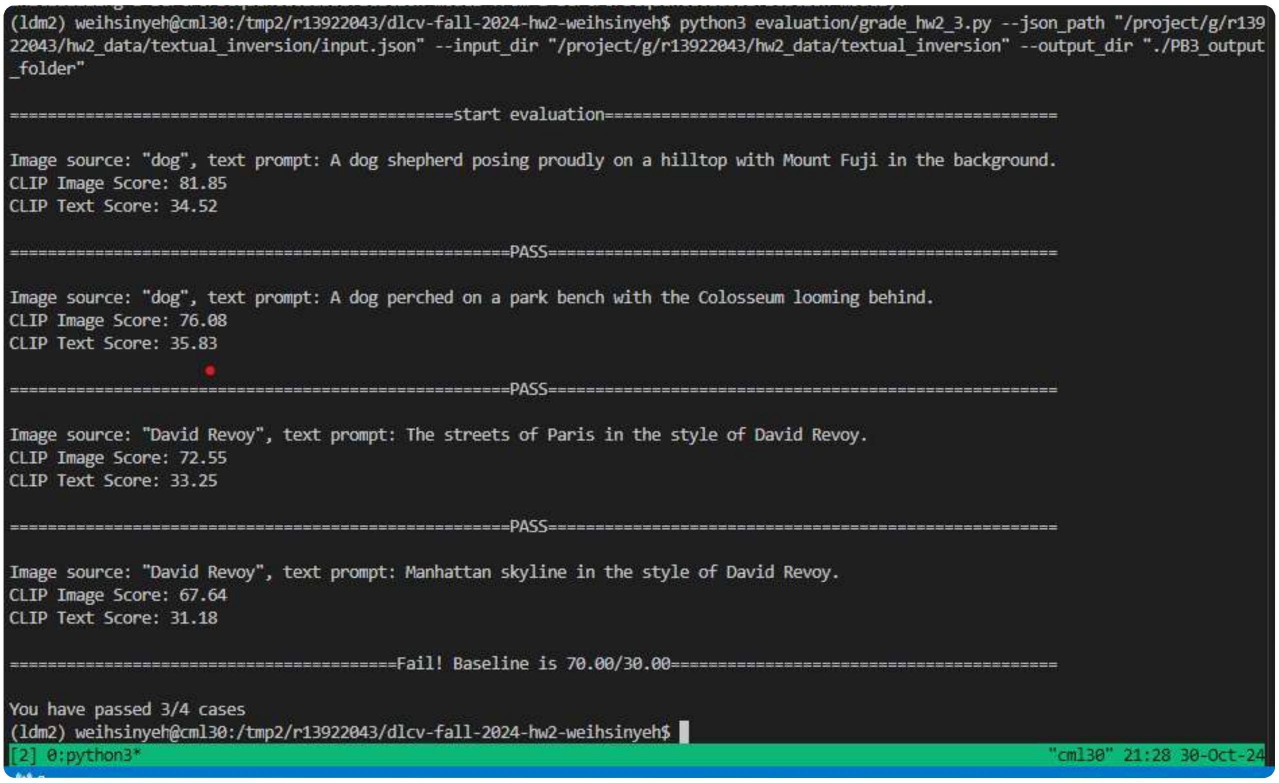


實驗中可以看到線性插值中，從左到右的影像慢慢變成粉紅色，同時也不像最右邊的女生。我上網查詢這種效果可能是來自於線性插值在潛在空間中對每個維度直接插值，卻沒有考慮到潛在空間的曲率，導致在中間的插值，會產生缺乏連續性，在現實中不存在的影像。而 slerp 在插值的過程中，因為在 latent space 中維持一樣的距離。所以避免了這樣直接插值失真的結果

Problem 3: Personalization (35%) [personalization dataset]

> In this problem, you need to implement Textual Inversion (https://arxiv.org/abs/2208.01618) on different concepts.

Evaluation



```
(ldm2) weihsinyeh@cml30:/tmp2/r13922043/dlcv-fall-2024-hw2-weihsinyeh$ python3 evaluation/grade_hw2_3.py --json_path "/project/g/r139
22043/hw2_data/textual_inversion/input.json" --input_dir "/project/g/r13922043/hw2_data/textual_inversion" --output_dir "./PB3_output
_folder"

===============================================start evaluation===============================================

Image source: "dog", text prompt: A dog shepherd posing proudly on a hilltop with Mount Fuji in the background.
CLIP Image Score: 81.85
CLIP Text Score: 34.52

=====================================================PASS=====================================================

Image source: "dog", text prompt: A dog perched on a park bench with the Colosseum looming behind.
CLIP Image Score: 76.08
CLIP Text Score: 35.83

=====================================================PASS=====================================================

Image source: "David Revoy", text prompt: The streets of Paris in the style of David Revoy.
CLIP Image Score: 72.55
CLIP Text Score: 33.25

=====================================================PASS=====================================================

Image source: "David Revoy", text prompt: Manhattan skyline in the style of David Revoy.
CLIP Image Score: 67.64
CLIP Text Score: 31.18

=========================================Fail! Baseline is 70.00/30.00=========================================

You have passed 3/4 cases
(ldm2) weihsinyeh@cml30:/tmp2/r13922043/dlcv-fall-2024-hw2-weihsinyeh$
[2] 0:python3*                                                                              "cml30" 21:28 30-Oct-24
```

Image source: "dog", text prompt: A dog shepherd posing proudly on a hilltop with Mount Fuji in the background.
CLIP Image Score: 81.85
CLIP Text Score: 34.52

PASS

Image source: "dog", text prompt: A dog perched on a park bench with the Colosseum looming behind.
CLIP Image Score: 76.08
CLIP Text Score: 35.83

PASS

Image source: "David Revoy", text prompt: The streets of Paris in the style of David Revoy.
CLIP Image Score: 72.55
CLIP Text Score: 33.25

PASS

Image source: "David Revoy", text prompt: Manhattan skyline in the style of David Revoy.
CLIP Image Score: 67.64
CLIP Text Score: 31.18

Fail! Baseline is 70.00/30.00

You have passed 3/4 cases

Report

(7.5%) 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.

Use the prompt “A photo of {object}.”
The naming rules of Images are {class_id}_{image_id}.png.
The id-to-label mapping is provided in hw2_data/clip_zeroshot/id2label.json
You should be able to import clip after installing the package from environment.yaml.
You can follow the sample code in CLIP repo.
程式碼使用 sample

```
Accuracy: 56.56%
```

```
5 Successful Cases:
Image: 17_464.png, True Label: kangaroo, Predicted Label: kangaroo
Image: 39_459.png, True Label: palm_tree, Predicted Label: palm_tree
Image: 49_456.png, True Label: sunflower, Predicted Label: sunflower
Image: 17_452.png, True Label: kangaroo, Predicted Label: kangaroo
Image: 44_465.png, True Label: elephant, Predicted Label: elephant
```

17_464.png

39_459.png

49_456.png

17_452.png

44_465.png

```
5 Failed Cases:
Image: 15_478.png, True Label: raccoon, Predicted Label: lamp
Image: 6_473.png, True Label: forest, Predicted Label: pine_tree
Image: 13_452.png, True Label: willow_tree, Predicted Label: oak_tree
Image: 48_456.png, True Label: shrew, Predicted Label: lizard
Image: 26_491.png, True Label: rose, Predicted Label: willow_tree
```

15_478.png

6_473.png

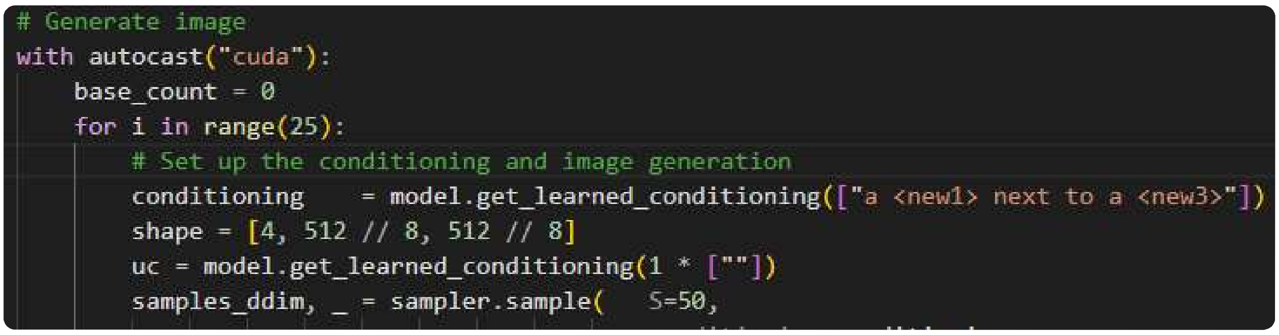
13_452.png

48_456.png

26_491.png

(7.5%) 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.

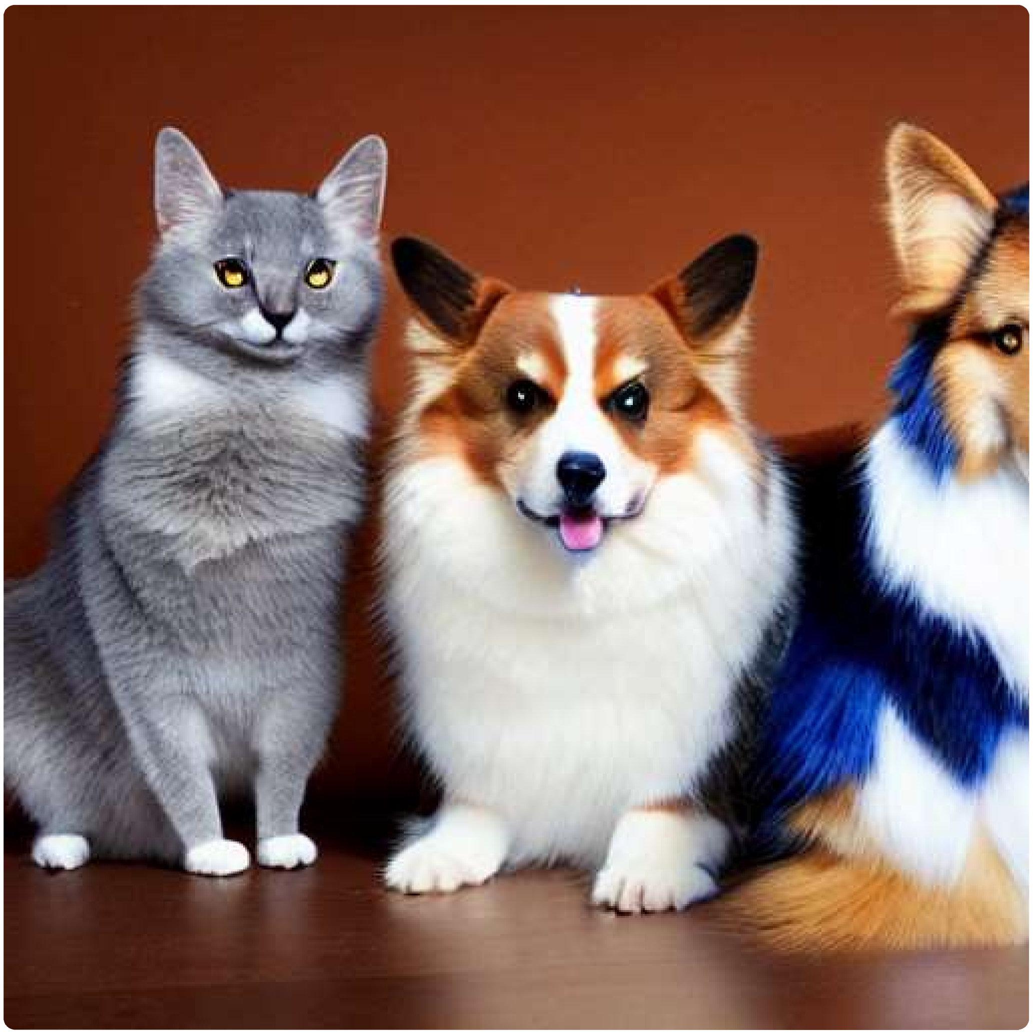
我使用助教提供的 cat images 來訓練新的 embedding `<new3>`



```
# Generate image
with autocast("cuda"):
    base_count = 0
    for i in range(25):
        # Set up the conditioning and image generation
        conditioning    = model.get_learned_conditioning(["a <new1> next to a <new3>"])
        shape = [4, 512 // 8, 512 // 8]
        uc = model.get_learned_conditioning(1 * [""])
        samples_ddim, _ = sampler.sample(   S=50,
```

會看到兩隻像柯基的，但是左邊那一隻是灰色的，因為助教提供的 cat images 都是灰色的貓。或是有混顏色的柯基出現，有學習到貓的風格，但卻不太像。有些也會出現奇怪的不像狗也不像貓，這些 concepts 混在一起的情況發生。有concept 混淆的情形。

偶爾也有成功的照片，如下。但仍然有一點不像真的。



Share your findings and survey a related paper that works on multiple concepts personalization, and share their method.

OMG: Occlusion-friendly Personalized Multi-concept Generation in Diffusion Models (https://arxiv.org/pdf/2403.10983)

這篇論文提出一個 two-stage multi-concept customization framework 。這個架構會整合多種 concepts 到一張圖片。這篇論文提出的方法在論文中說可以解決 identity degradation, occlusion, time-consuming fusion, and illumination disharmony problems. 這次在做作業中，將柯基這個 concept 加入到新的 text 後生出的照片的柯基會變得長的很奇怪。讓畫面非常不協調。有時是柯基的舌頭不在對的地方，或是身體扭曲。而這篇論文的架構圖如下:

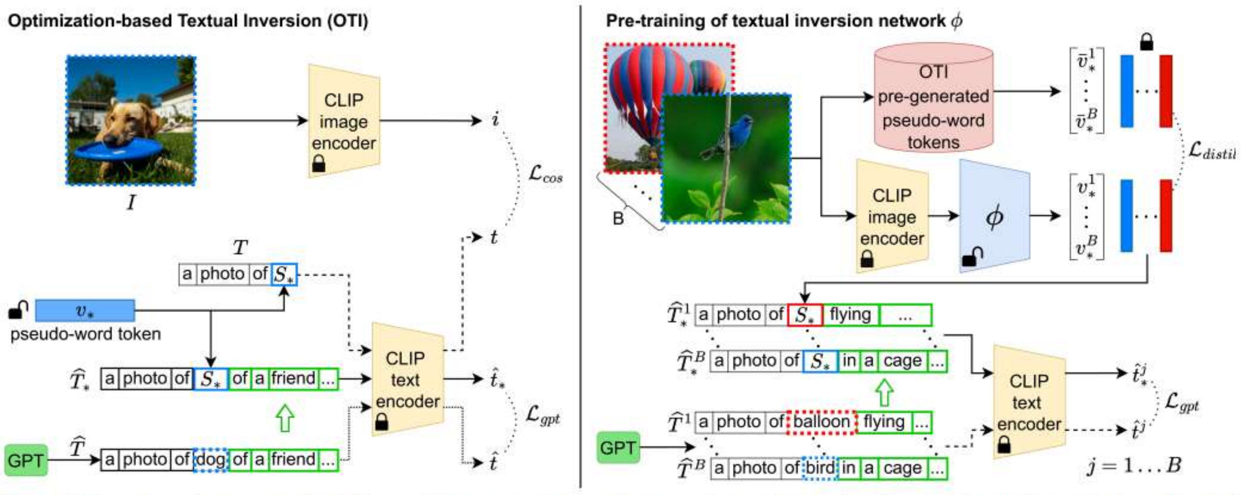

Optimization-based Textual Inversion (OTI)
CLIP image encoder
I
i
$\mathcal{L}_{cos}$
T
a photo of S_*
t
v_*
pseudo-word token
$\hat{T}_*$ a photo of S_* of a friend ...
CLIP text encoder
$\hat{t}_*$
$\mathcal{L}_{gpt}$
GPT
$\hat{T}$ a photo of dog of a friend ...
$\hat{t}$
Pre-training of textual inversion network ϕ
OTI pre-generated pseudo-word tokens
B
CLIP image encoder
ϕ
$\bar{v}_*^1$ $\bar{v}_*^B$
v_*^1 v_*^B
$\mathcal{L}_{distil}$
$\hat{T}_*^1$ a photo of S_* flying ...
$\hat{T}_*^B$ a photo of S_* in a cage ...
$\hat{T}^1$ a photo of balloon flying ...
$\hat{T}^B$ a photo of bird in a cage ...
CLIP text encoder
$\hat{t}_*^j$
$\hat{t}^j$
$\mathcal{L}_{gpt}$
$j = 1 \dots B$

Figure 2. Overview of our approach. *Left*: we generate a pseudo-word token v_* from an image I with an iterative optimization-based textual inversion. We force v_* to represent the content of the image with a cosine loss $\mathcal{L}_{cos}$. We assign a concept word to I with a CLIP zero-shot classification and feed the prompt "a photo of {concept}" to GPT to continue the phrase, resulting in $\widehat{T}$. Let S_* be the pseudo-word associated with v_*, we build $\widehat{T}_*$ by replacing in $\widehat{T}$ the concept with S_*. $\widehat{T}$ and $\widehat{T}_*$ are then employed for a contextualized regularization with $\mathcal{L}_{gpt}$. *Right*: we train a textual inversion network ϕ on unlabeled images. Given a set of pseudo-word tokens pre-generated with OTI, we distill their knowledge to ϕ through a contrastive loss $\mathcal{L}_{distil}$. We regularize the output of ϕ with the same GPT-powered loss $\mathcal{L}_{gpt}$ employed in OTI. B represents the number of images in a batch.

此論文的方法第一階段 Visual Comprehension Information Preparation :
第一階段旨在為多概念定制獲取視覺理解資訊。傳統方法如 Mix-of-show 方法在佈局衝突方面存在挑戰，當兩個概念的區域互相遮擋時，該方法無法產生具有一致性佈局的圖像，導致概念的完整性受損，這個問題叫 layout conflict。

有鑒於 cross-attention layers 在控制 spatial layout and appearance 很有效果，以及在每層 layer 用以 pixel-to-text 互動的方式讓模型可以保存原先圖片的內容以及空間的資訊，同時趨近於目標的 prompt 。
互動的方式是透過在 layer 中選擇修改影像的預定區域，保留影像的內容和結構。在這個階段，Text to Image 模型接受類別名稱（如「man」或「woman」）的 prompt，以保留整體的佈局。用這樣來解決 identity degradation。

此論文的方法第二階段 Multi-concept Personalized Denoising 透過注入概念的身份特徵來生成特定圖像。此外還提出了（Concept Noise Blending），在推理過程中利用多個單一概念模型生成多概念圖像。此方法不僅減少additional optimization costs associated with network merging,
額外的優化成本在合併不同 network 中，在推理過程中直接利用多個單一概念模型，避開了網路合併的需求。此外，每個單一概念模型僅負責生成特定概念，有效解決了身份劣化的問題，確保每個概念的清晰呈現。

Reference

Hugging face : textual_inversion/textual_inversion.py (https://github.com/huggingface/diffusers/blob/main/examples/textual_inversion/textual_inversion.py)
Textual-Inversion (https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Textual-Inversion)