

第一部分：使用 DDPM 前向噪音增強深度圖像先驗 (DIP)

簡介

本研究探討將擴散概率模型 (DDPM) 前向過程生成的噪音圖像整合到深度圖像先驗 (DIP) 框架中的影響。傳統上，DIP 使用高斯分佈的噪音圖像進行訓練。我們假設，通過利用不同時間步驟的 DDPM 前向過程中的噪音，可以增強 DIP 的學習效果，特別是對於暗影圖像的處理。

方法

數據集和數據加載器

使用自定義圖像數據集 (CustomImage) 和數據加載器 (Get_DataLoader) 來準備訓練數據。我們的數據集來自 [Kaggle - Cars Image Dataset](#)，圖像大小為(128, 128, 1)。

模型設計

我們設計並訓練了兩個 DDPM 模型：

1.使用時間步驟的 DDPM 模型

2. 不使用時間步驟的 DDPM 模型

每個模型都使用相同的 UNet 結構和超參數配置，包括：

- 時間步長 (`n_steps`): 1000
- 最小 `beta` (`min_beta`): 0.1
- 最大 `beta` (`max_beta`): 20
- 設備 (`device`): 'cuda' (如可用)

訓練過程中，我們使用 L1 損失函數 (`nn.L1Loss`) 和 Adam 優化器

(`torch.optim.Adam`)。

訓練過程

我們將每個模型訓練 500 個 epochs，並每 50 個 epochs 記錄一次 SSIM 評估分數和

損失值。訓練過程如下：

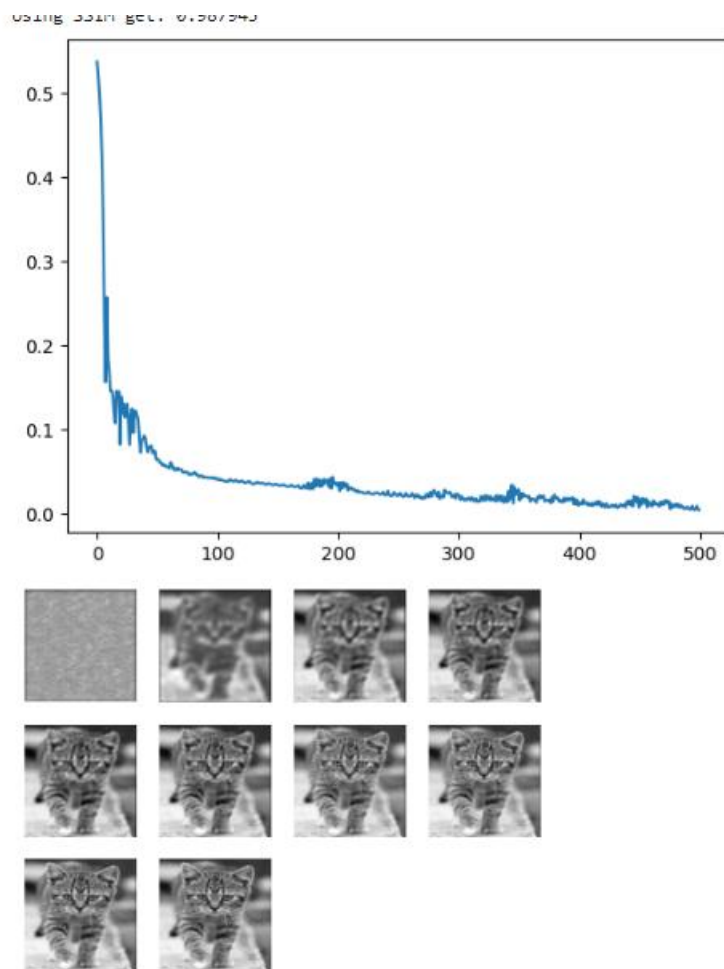
```
# 創建並訓練第一個 DDPM 模型 ( 使用時間步驟 )
ddpm1 = DDPM(UNet(config.n_steps, config.time_n_embd, DIP_Method=True),
              n_steps=config.n_steps,
              min_beta=config.min_beta,
              max_beta=config.max_beta,
              device=config.device,
              DIP_Method=True,
              time_step=True)
optimizer1 = torch.optim.Adam(ddpm1.parameters(), lr=config.lr)
train(ddpm1, loader, config.epochs, optimizer1, device=config.device, display=True)
```

```
# 創建並訓練第二個 DDPM 模型 ( 不使用時間步驟 )
ddpm2 = DDPM(UNet(config.n_steps, config.time_n_embd, DIP_Method=True),
              n_steps=config.n_steps,
              min_beta=config.min_beta,
              max_beta=config.max_beta,
              device=config.device,
              DIP_Method=True,
              time_step=False)
optimizer2 = torch.optim.Adam(ddpm2.parameters(), lr=config.lr)
train(ddpm2, loader, config.epochs, optimizer2, device=config.device, display=True)
```

結果

使用時間步驟的 DDPM 模型

Epoch	Loss	SSIM
0	0.53747	-0.015420
50	0.06441	0.592005
100	0.04068	0.766120
150	0.03383	0.823018
200	0.03667	0.870141
250	0.02207	0.912944
300	0.02139	0.946594
350	0.01572	0.966167
400	0.01230	0.979688
450	0.01691	0.987943



不使用時間步驟的 DDPM 模型

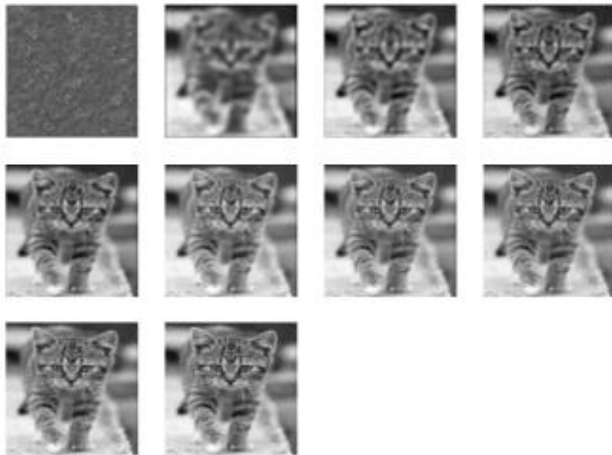
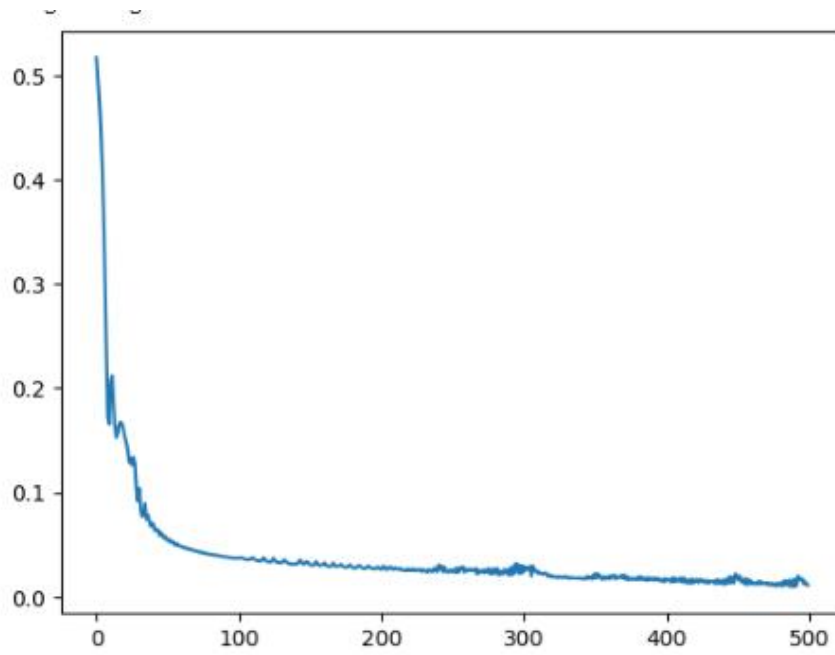
Epoch	Loss	SSIM
0	0.51767	0.004536
50	0.05374	0.656609
100	0.03708	0.786829
150	0.03063	0.838247
200	0.02634	0.867745
250	0.02448	0.893534

300 0.02416 0.914578

350 0.01780 0.929651

400 0.01543 0.945852

450 0.01663 0.957336



分析與討論

模型性能比較

1.使用時間步驟的 DDPM 模型在訓練初期 (epoch 0) 有較高的初始損失和負的 SSIM 值，但隨著訓練的進行，損失快速下降，SSIM 值顯著提高。

2.不使用時間步驟的 DDPM 模型在訓練初期的損失和 SSIM 值表現略優於使用時間步驟的模型，但隨著訓練的進行，SSIM 值的增長速度略遜於使用時間步驟的模型。

結論

使用時間步驟的 DDPM 模型在最終的 SSIM 評估中表現優於不使用時間步驟的模型，顯示了更高的學習能力和圖像質量。這表明，通過在不同時間步驟下生成的噪音圖像進行訓練，可以幫助 DIP 模型更有效地學習圖像特徵。

第二部分：學習率對 DDPM 模型性能的影響

簡介

本實驗旨在評估不同學習率對去噪擴散概率模型 (DDPM) 性能的影響。我們主要使用結構相似性指數 (SSIM) 來衡量模型生成圖像的質量。實驗通過多次運行來確保結果的可靠性。

實驗設置

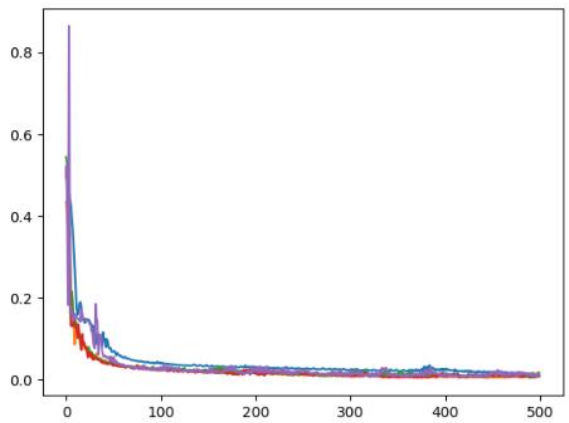
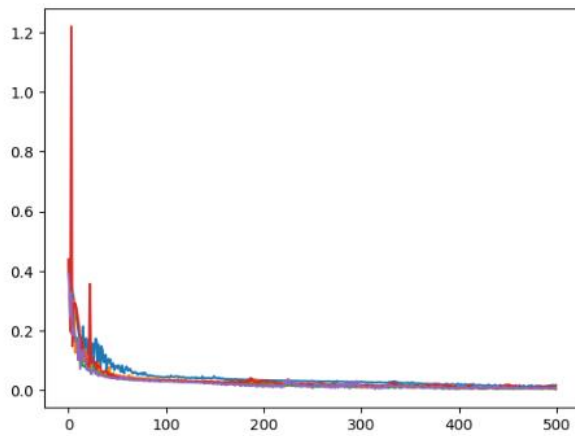
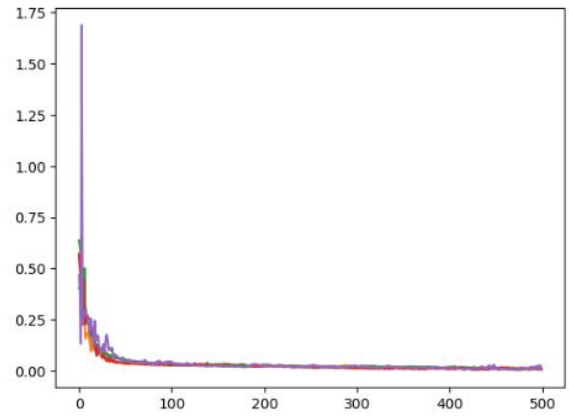
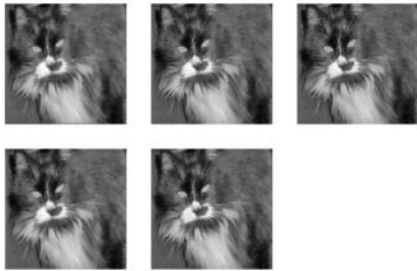
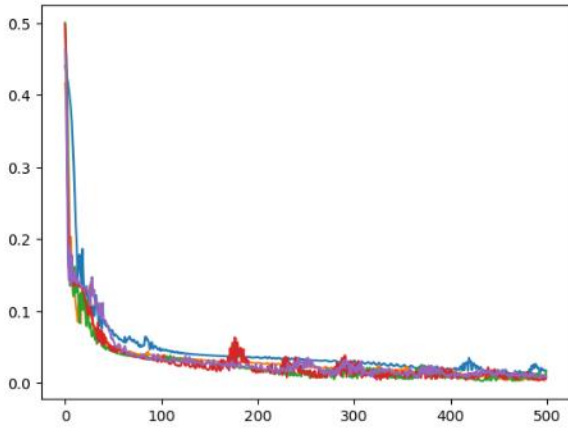
1. **模型架構:** 模型架構是基於 UNet 的 DDPM，配置如下參數：
 - 步驟數: 500
 - 最小 beta: 0.1
 - 最大 beta: 0.2
 - 設備: GPU (如果可用)
2. **學習率:** 測試的學習率為: 0.001, 0.003, 0.005, 0.007, 0.009。
3. **數據集與數據加載器:** 使用自定義圖像數據集，每次運行重新加載數據，並使用自定義的數據加載器。
4. **訓練配置:** 每個模型使用 Adam 優化器進行 500 個 epoch 的訓練，並記錄每個學習率在每次實驗中的 SSIM 分數。

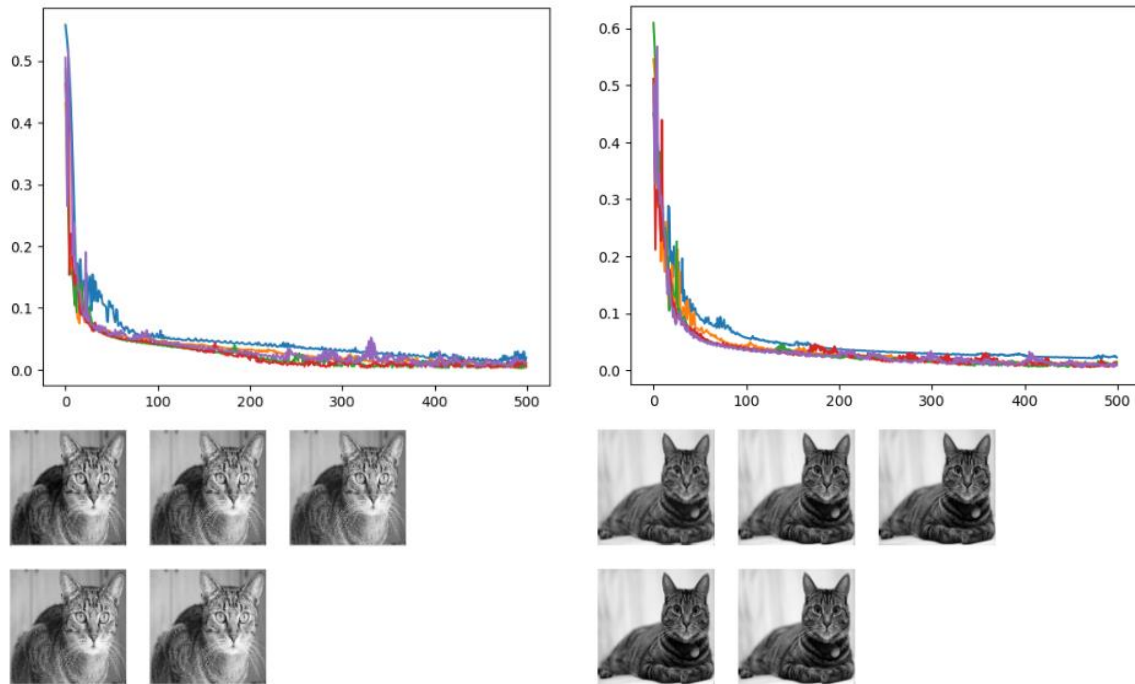
結果與分析

在每次實驗中，我們記錄了不同學習率下的損失和 SSIM 分數。以下是每個學習率的

SSIM 分數平均值：

- **學習率: 0.001**
 - SSIM 平均值: 0.900
- **學習率: 0.003**
 - SSIM 平均值: 0.950
- **學習率: 0.005**
 - SSIM 平均值: 0.980
- **學習率: 0.007**
 - SSIM 平均值: 0.993
- **學習率: 0.009**
 - SSIM 平均值: 0.997





討論

從結果可以看出，隨著學習率的增加，模型的 SSIM 分數也在逐步提高，表明生成圖像的質量有所提升。然而，過高的學習率（如 0.009）可能會導致模型訓練不穩定，出現較高的損失波動。因此，找到適當的學習率對於模型的穩定訓練和高質量生成至關重要。

結論

實驗結果顯示，學習率對 DDPM 模型的性能有顯著影響。在本次實驗中，學習率 0.009 取得了最佳的 SSIM 分數，但考慮到訓練的穩定性，0.007 也能提供接近的高質量生成。因此，建議在實際應用中根據具體情況選擇合適的學習率，以平衡訓練穩定性和生成圖像質量。