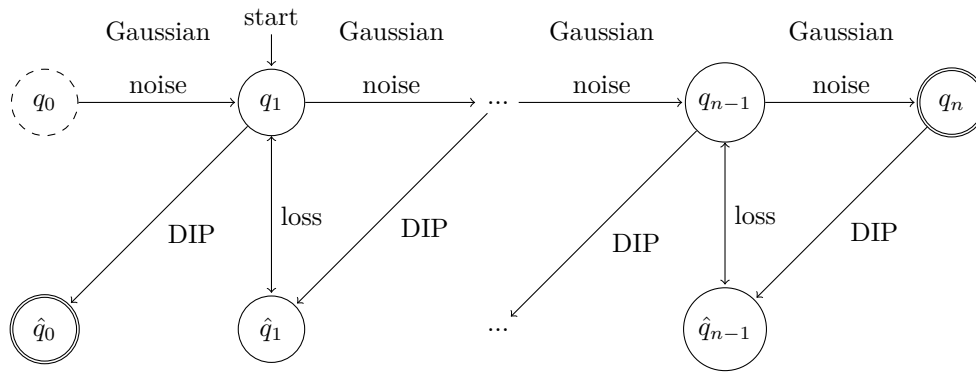


GitHub Link: https://github.com/EdwinTsai1231/GAI_project4

1 Theoretical Justification

1.1 使用 DDPM 的觀念輔助 DIP 的訓練過程

目標: 利用 DDPM 在原始圖片 (q_0) 上逐步增加 Gaussian noise 使得圖片可視為純粹的 Gaussian noise (q_n) 的特性讓 DIP 模型從反向逐步學習 denoising 生成接近原始無雜訊的圖片。



- q_0 : 為原始無雜訊的圖片，在應用場景通常是無法取得的
- q_1 : 可取得的 (最小) 雜訊圖片
- q_k : 為在 q_{k-1} 上加 Gaussian noise 的圖片, 其中 $0 < k \leq n$
- q_n : 為受 Gaussian noise 干擾最嚴重的圖片, 根據 DDPM 的原理此 q_n 可視為一個純粹的 Gaussian noise
- \hat{q}_{n-1} : 為將 Gaussian distribution 的雜訊 q_n 輸入至 DIP 得到的結果
- \hat{q}_k : 為將 q_{k+1} 輸入至 DIP 得到的結果, 其中 $0 \leq k < n - 1$
- \hat{q}_0 : 為接近原始無雜訊圖片的結果

1.2 將 DDPM 以及 DIP 模型結合的預期效益

藉由 DDPM 將原始圖片逐漸增加 Gaussian noise 使得圖片最終變成純粹 Gaussian distribution 的概念，我們可以讓 DIP 模型利用前後有無增加 Gaussian noise 的圖片 (e.g., q_{k-1} , q_k) 訓練出一個能夠 denoise 的模型。在訓練的時候最一開始可以輸入 Gaussian distribution 的雜訊 (q_n)，接著由於

DDPM 擁有添加 Gaussian noise 前的輸入圖片 (q_k) 且 DIP 模型只需要單筆資料就可以訓練的特性，此圖片便可以當做是 DIP 模型的 target，讓 DIP 模型輸出的圖片 (\hat{q}_k) 能夠計算 $loss(q_k, \hat{q}_k)$ 的值，其中 $loss$ 代表 loss function。

透過此訓練過程的 DIP 模型預期可以學習該圖片的階層式去噪過程，且由於預期該模型學會如何去雜訊因此將 q_1 (已知擁有的雜訊圖片) 輸入該模型時便可以得到一個不錯的去雜訊的圖片 \hat{q}_0 (預期比原本只使用 DIP 模型的效果好)。

1.3 將 DDPM 以及 DIP 模型結合的限制

相比只使用 DIP 模型不添加 DDPM 連續 Gaussian noise 的方法，該方法有幾個缺點:

- 由於 DDPM 的特性，可取得的圖片 (q_1) 需假設是受到 Gaussian noise 影響，若為 salt-and-pepper noise 可能就無法適用。
- 由於需要計算多組 (q_k, \hat{q}_k) 圖片的 loss 值，其所需的 GPU 記憶體空間 相較只須計算一組的 DIP 模型來的大。

2 Experimental Verification

為了驗證該理論，我會以以下的訓練演算法對我的 Hierarchical DIP 模型進行訓練

Algorithm 1: Hierarchical DIP

Input: q_1 : a image with Gaussian noise

Output: \hat{q}_0 : the reconstruction image

```

1  $q_1..q_n \leftarrow create\_noisy\_images(q_1);$ 
2  $e \leftarrow 0;$ 
3 while  $e < num\_epoch$  do
4   foreach  $i$  in  $n - 1..1$  do
5      $\hat{q}_i \leftarrow DIP(q_{i+1});$ 
6      $loss\_value \leftarrow loss(q_i, \hat{q}_i);$ 
7      $update(DIP, loss\_value)$ 
8   end
9    $e \leftarrow e + 1;$ 
10 end
11  $\hat{q}_0 \leftarrow DIP(q_1);$ 
12 return  $\hat{q}_0;$ 
```

- Step 1. 以 q_1 圖片為基底，逐步增加 Gaussian noise e.g., q_k 的圖片就是以 q_{k-1} 基礎之上在增加 Gaussian noise, 直到 q_n , 其中 n 為一個 hyperparameter 可以進行調整。
- Step 2-10. 為模型訓練過程。
- Step 4-8. 從雜訊最嚴重到最輕微的圖片，依次輸入至模型，並計算該輸出 (\hat{q}_k) 與在產生雜訊圖片的過程裡相對應圖片 (q_k) 的 $loss_value$ ，並根據該 $loss_value$ 為依據更新模型參數。
- Step 11-12. 將原始圖片 (雜訊最小的圖片, q_1) 輸入至該模型，並得出一個接近原始理想無雜訊的圖片 \hat{q}_0 ，並回傳該結果。

2.1 Standalone DIP 模型生成的效果

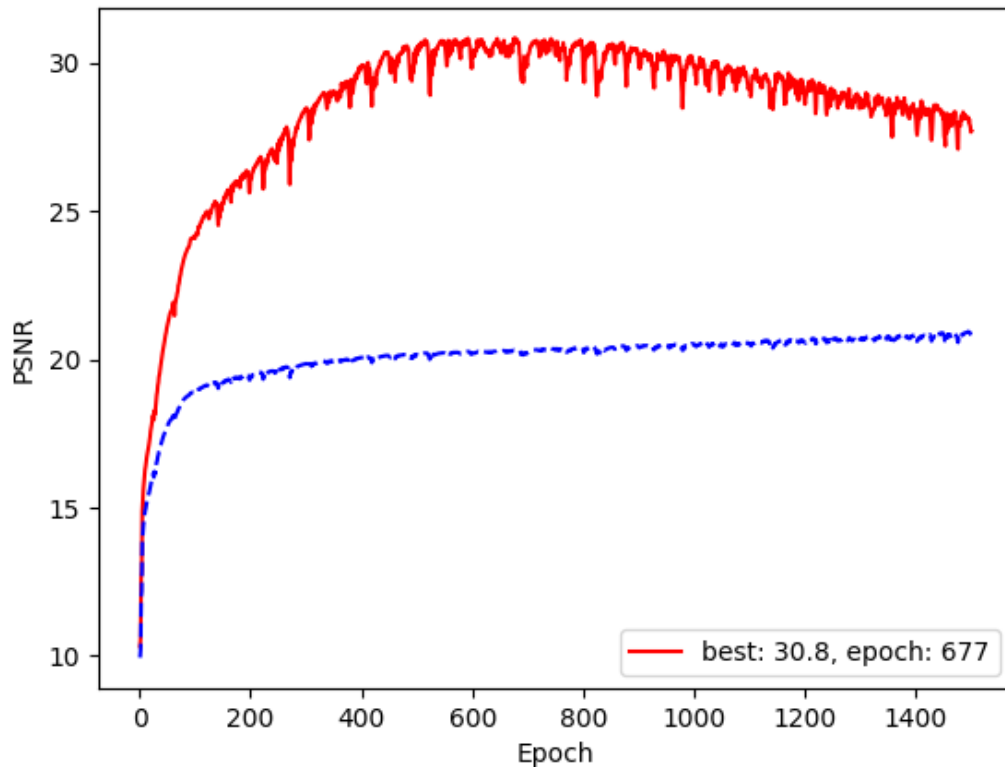


Figure 1: 只使用 DIP 所得出的結果; 藍色虛線為 DIP 的輸出對 noise image 計算出來的 PSNR, 紅色實線為 DIP 的輸出對無雜訊圖片計算出來的 PSNR

可以從 Figure 2得知，由於 DIP 只須跟 noise image 計算 loss function 並更新參數，因此藍色的虛線會逐漸上升。同時 DIP 在訓練初期會著重在生成圖像中 low frequency 的部份, high frequency 的

部份 e.g., noise 則會在訓練後期才會去學習。因此紅色實線 (DIP 生成的圖片與無雜訊圖片的 PSNR) 會是先上升 (學習圖像 low frequency 部份, 但尚未學習到 high frequency) 後下降 (學習到包括 noise 在內的 high frequency 部份)。在 1000 次 epoch 的條件下, 在第 677 次 epoch 情況下我的 DIP 生成圖片與無雜訊圖片的 PSNR 到達最高數值 30.8。

2.2 Hierarchical DIP 模型 (DDPM + DIP) 生成的效果

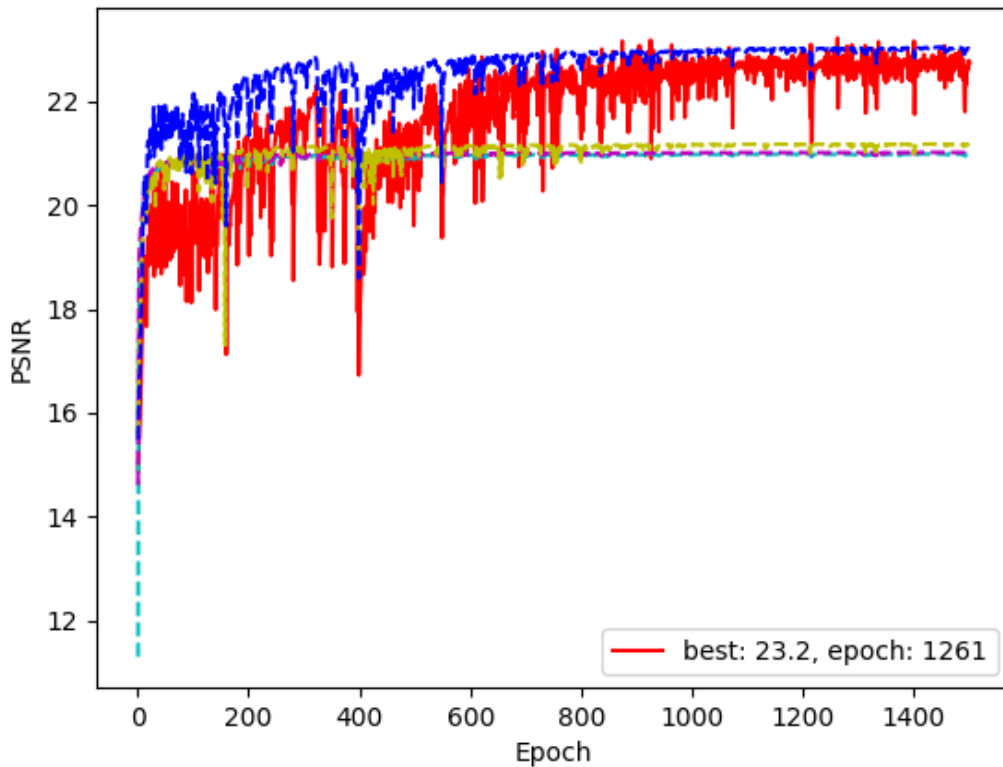


Figure 2: 藍色虛線為 DIP 的輸出對 noise image 計算出來的 PSNR, 紅色實線為 DIP 的輸出對無雜訊圖片計算出來的 PSNR, 青色虛線、洋紅色虛線、黃色虛線皆為在中間過程中模型產生的圖與相對應的雜訊圖之間的 PSNR 值

在 1500 epoch 的情況下, 使用 Hierarchical DIP 模型 (DDPM + DIP) 生成最好的效果出現在第 1261 次, 其 PSNR 為 23.2

2.3 Visual Quality of the Images

Figure 5 以及 Figure 6 分別為使用 Standalone DIP 模型以及 Hierarchical DIP 模型訓練期間生成 PSNR 結果最好的圖。為了比較方便也展示了 ground truth ($q_0, 3$) 以及首張雜訊圖 ($q_1, 4$)。

Figure 3: Ground Truth q_0 Figure 4: 雜訊圖片 q_1 

Figure 5: 只使用 DIP 模型 PSNR 最好的圖片



Figure 6: 使用 hierarchical DIP 模型 PSNR 最好的圖片

2.4 Conclusion

無論從 PSNR 對 EPOCH 作圖的結果或是從最終圖片生成出來的品質，我所提出的 Hierarchical DIP 模型在去雜訊的品質以及訓練該模型所需的時間效率皆不如 Standalone DIP。但是 Hierarchical DIP 模型仍有一個優點，即是此模型最終圖片生成的品質 (PSNR, Figure 2 紅色實線) 與最後第二張雜訊圖 q_2 去噪成最後一張 \hat{q}_1 的 PSNR 值 (Figure 2 藍色虛線) 相關，因此不需要考慮需要 early stopping 的時間點。