

GAI Project 4 Report

Generative Models for Visual Signals

資訊 114 F7401254 張曄俊

1. **GitHub Link:** <https://github.com/PDrookie/Generative-Models-for-Visual-Signals>

2. 前言

本次選擇的是 Example 2 作為實作的目標，會選擇第 2 個而非第 1 個 Example 的原因是因為在與同學討論以及更了解 DIP 和 DDPM 後，相較於第二個 Example 是改良 DIP 模型，使 DDPM 藉由 DIP 模型做加速，我認為可能相對較難以實現，屏除 DIP 在我試驗時，感覺速度相對不快以外，以 DDPM 的原理來看，其受 DIP 模型的影響應該不夠大到在改良後有明顯的效果，可能實作上相對困難，因此我選擇 Example 2 來實作。

3. 與傳統 DIP 模型的比較

在本次的模型架構上，皆是採用簡單的 CNN 基礎架構。

```
class DIPModel(nn.Module):
    def __init__(self):
        super(DIPModel, self).__init__()
        self.conv1 = nn.Conv2d(3, 64, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(64, 128, kernel_size=3, padding=1)
        self.conv3 = nn.Conv2d(128, 64, kernel_size=3, padding=1)
        self.conv4 = nn.Conv2d(64, 3, kernel_size=3, padding=1)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.relu(self.conv2(x))
        x = self.relu(self.conv3(x))
        x = self.conv4(x)
        return x
```

(圖一：DIP 模型架構示意圖)

本次在訓練時使用的 learning rate 皆為 0.05，總共執行的 epochs 數皆為 3000。另外在對於 early stopping method 的設計上，是以 PSNR 的評測分數為判斷依據，每執行 10 epochs 後判斷一次，使用一個名為 patience 的參數作為停止標準，當我們當今的 PSNR 分數與過往訓練出來最佳的 PSNR 分數相差小於 0.001 的話，我們就對 no_improvement_count 這個參數加一，一旦 no_improvement_count 的數值大於 patience 的數值，就立即停止訓練動作，並輸出最終結果，而在本次所有的實驗都是將 patience 設定為 50，也就是一旦超過 500 個 epochs 我們的 PSNR 未有顯著變動，我們即停止訓練。

```
# Check for early stopping
if current_psnr - best_psnr >= 0.001:
    best_psnr = current_psnr
    best_model = model.state_dict()
    # no_improvement_count = 0 # Reset counter if improvement
else:
    no_improvement_count += 1

if no_improvement_count >= patience:
    print(f"Early stopping at epoch {epoch}")
    break
```

(圖二：early stopping method 示意圖)

在傳統的 DIP 模型，我所使用的 noise level 設定為 0.1，而在改良後的 DIP 模型我使用的 level 則分三個階段，也就是 DDPM 啟發的監督信息來輔助 DIP 判斷 early stopping。分為一開始的 0.1，接著 0.05 以及最後 0.025，設計上是以每一階段除以二。訓練的 epochs 數則是將 3000 平分給 3 個階段，也就是每個階段訓練 1000 個 epochs。

評測前根據所知道的理論來說，應該會是改良後的結果表現相對的較為優良，因為相對傳統以 CNN 簡單的架構而言，改良後的 DIP 模型有了 DDPM 的設計框架，可以相對於傳統 DIP 模型更好地找到早停點，使得我們圖片呈現的結果更好。

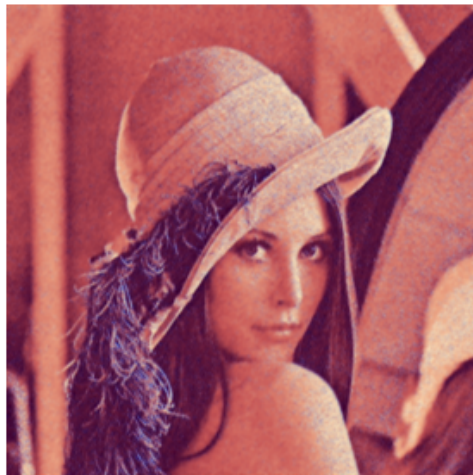
評測結果如下表所示：

(表一：傳統 DIP 模型與改良 DIP 模型比較表)

| | Traditional | Modified |
|----------------------|---------------|-------------------------|
| Costing Time | 118.4s | 251.4s |
| PSNR Score | 26.2294 | 34.5017 |
| SSIM Score | 0.7426 | 0.9038 |
| Early Stopping state | At Epoch 1370 | At Epoch 680 in Stage 3 |

從結果可以發現，除了分數的表現都相對優良以外，我們 early stopping 的時間點，也相對於一般傳統的 DIP 模型來的更後面一些，也因此所耗費的時間較長一些，而這些也都符合我們理論所想。

DIP Output



(圖三 (左)：傳統 DIP 模型的圖片最後成果)

DIP Output



(圖四 (右)：改良後的 DIP 模型的最後圖片成果)

從圖三圖四也能看出在 Renoise (DIP Output) 之後，我們傳統的結果仍相對地有些粗糙，改良後的結果則可以說是逼近原圖。

4. 尋找最佳的改良參數設定

A. 改變模型架構的層數

在設計整個 DIP 模型時，我先入為主的認為模型應該要有一定深度表現會相對來的好，因此我選擇將 DIP 模型的層數在一開始設定為四層，但根據作業說明的建議，我還是實作了當模型層數不同時，對評測結果的影響，但以我學習到的理論來說，我認為層度越大，表現結果也會相對的較好。

評測結果如下表：

(表二：改良 DIP 模型在模型層度不同的比較表)

| | 3 Layers | 4 Layers | 5 Layers |
|-----------------------------|----------------------------------------------------------|----------------------------|----------------------------------------------------------|
| Costing Time | 99.1s | 251.4s | 486.1s |
| PSNR Score | 34.9749 | 34.5017 | 30.1730 |
| SSIM Score | 0.9205 | 0.9038 | 0.8500 |
| Early Stopping state | At Epoch 640 in Stage 2 At Epoch 680 in Stage 3 | At Epoch 820 in Stage 3 | At Epoch 980 in Stage 2 At Epoch 580 in Stage 3 |

最後實驗出來的結果卻跟我所想的不盡相同，可以發現，儘管 4 層與 3 層的表現結果可以說相差無幾，但總體來說還是 3 層的模型表現的相對較好。為了找尋會這樣的原因，在查了一些資料後，得出來的結論是，可能是因為圖片太過簡單，不需要過複雜的模型與訓練過程，所以才會使得簡單的模型有相對較好的訓練成果。

有趣的是，在這邊可以發現 early stopping method 也開始發揮作用，當訓練分數趨於收斂時，就會停止目前這個階段的訓練，因此能提升效率，減少訓練時不必要的時間浪費。

另外，在傳統的 DIP 模型中，層數越低的表現也是相對較好，而且差距更加的明顯。

(表三：傳統 DIP 模型在模型層度不同的比較表)

| | 3 Layers | 4 Layers | 5 Layers |
|-----------------------------|---------------|---------------|---------------|
| Costing Time | 85.3s | 118.4s | 305.1s |
| PSNR Score | 31.0612 | 26.2294 | 25.9803 |
| SSIM Score | 0.8015 | 0.7426 | 0.7432 |
| Early Stopping state | At Epoch 1630 | At Epoch 1370 | At Epoch 1380 |

B. 在不同 level 變化快慢下的結果

接著我開始對 level 做設定上的調整，分為以下三種：

(表四：三種不同噪聲級別設定表)

| | Stage 1 | Stage 2 | Stage 3 |
|----------------|---------|---------|---------|
| Gradual | 0.1 | 0.05 | 0.025 |
| Fast | 0.1 | 0.01 | 0.001 |
| Slow | 0.1 | 0.9 | 0.8 |

Gradual 表示的是每往下一層除以 2，而 Fast 則是為了表示驟降的感覺每往下一層除以 10，最後 Slow 則是為了表現出緩慢下降或差異不大的感覺，每往下一層僅僅減去 0.1。

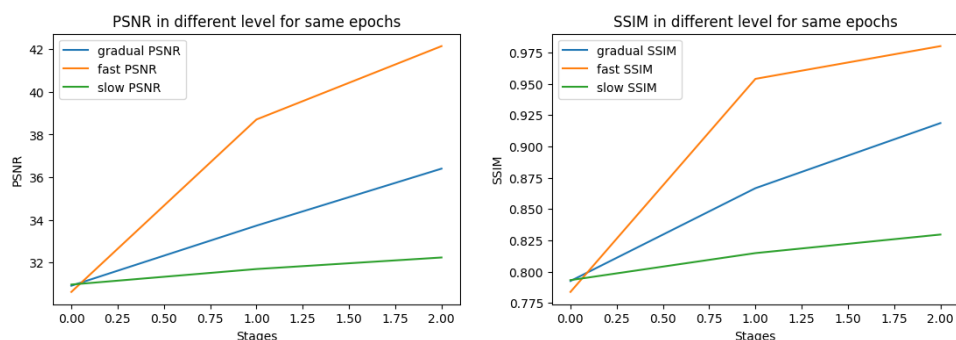
依據理論分析來說，Fast 可以再更低的噪聲級別獲得訓練，會在最終結果上有相對較好的表現，而 Slow 則是相反，因為相對的噪聲級別降得不夠多，可能訓練效果會比較不佳，而 Gradual 則應該會介於兩者之間，且會更偏向於 Slow，因為下降的幅度也不夠大。

訓練結果如下，每一級別皆使用 1000 epochs 做訓練，且因前述結果，使用 3 層 layers 建構 DIP 模型：

(表五：不同噪聲級別在改良後的 DIP 模型表現比較表)

(平均為每一個噪聲級別階段最後的評測分數平均)

| | Gradual | Fast | Slow |
|-----------------------------|----------------------------------------------------------|----------------------------|----------------------------------------------------------|
| PSNR mean Score | 33.6759 | 37.1566 | 31.6261 |
| SSIM mean Score | 0.9186 | 0.9800 | 0.8125 |
| Early Stopping state | At Epoch 700 in Stage 2 At Epoch 620 in Stage 3 | At Epoch 790 in Stage 3 | At Epoch 840 in Stage 2 At Epoch 680 in Stage 3 |



(圖五(左)：不同噪聲級別在改良後的 DIP 模型 PSNR 分數比較表)

(圖六(右)：不同噪聲級別在改良後的 DIP 模型 SSIM 分數比較表)

C. Epochs 數對於不同 level 變化的影響

在這個階段我延續了 B 階段使用的噪聲級別設定，也就是 Gradual、Fast、Slow 這三組，那這邊要實測的是當 epochs 數皆很小時，的影響，以及 epochs 數由小隨著級別階段增加一併增大的影響。

那在這邊根據前述 B 以及理論來推測，無論是哪一組在最後的表現上 Fast 應該一樣會是最好的，Gradual 次之，Slow 則是最後。

a. Epochs 數相對小

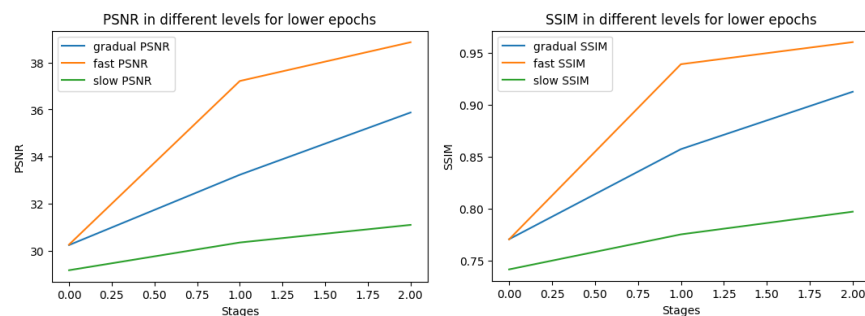
將 epochs 數目皆設定為 200，並比較其結果。

實測結果如下：

(表六：不同噪聲級別在相對較小的 epochs 數表現比較表)

(平均為每一個噪聲級別階段最後的評測分數平均)

| | Gradual | Fast | Slow |
|-----------------------------|---------|---------|---------|
| PSNR mean Score | 33.1197 | 35.4477 | 30.2149 |
| SSIM mean Score | 0.8467 | 0.8898 | 0.7711 |
| Early Stopping state | 完全執行 | 完全執行 | 完全執行 |



(圖七(左)：不同噪聲級別在在相對較小的 epochs 數 PSNR 分數比較表)

(圖八(右)：不同噪聲級別在在相對較小的 epochs 數 SSIM 分數比較表)

可以發現由於訓練的 epochs 數較小，所以 Slow 由於噪聲級別下降的不夠再加上訓練不足，導致整個表現都非常的糟糕，而 Gradual 也同理，但是 Fast 因為噪聲級別下降幅度夠大，使得她儘管在 epochs 數目相對小的情況也能有不錯的表現。

另外，可能因為 Epochs 的數量相對小不少，導致都還沒有完全訓練完成，所以並未出現提前暫停的情況。

b. Epochs 數隨著噪聲級別增加

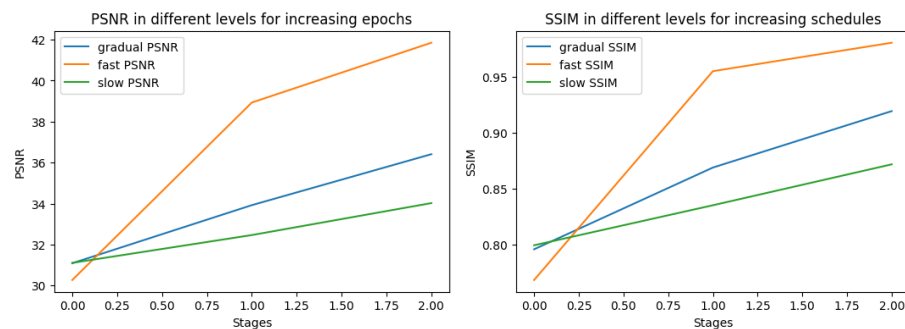
將 epochs 數目隨著噪聲級別階段的不同分別設定為，第一階段 500，第二階段 1000，第三階段 1500，總共 3000 epochs。

實測結果如下：

(表七：不同噪聲級別在 epochs 數隨之增加表現比較表)

(平均為每一個噪聲級別階段最後的評測分數平均)

| | Gradual | Fast | Slow |
|-----------------------------|--------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| PSNR mean Score | 33.1197 | 35.4477 | 30.2149 |
| SSIM mean Score | 0.8467 | 0.8898 | 0.7711 |
| Early Stopping state | At Epoch 670 in Stage 2 At Epoch 600 in Stage 3 | At Epoch 880 in Stage 2 At Epoch 590 in Stage 3 | At Epoch 580 in Stage 2 At Epoch 580 in Stage 3 |



(圖九(左)：不同噪聲級別在 epochs 數隨之增加 PSNR 分數比較表)

(圖十(右)：不同噪聲級別在 epochs 數隨之增加 SSIM 分數比較表)

由圖可以發現，因為一開始的 epochs 數都相對較少，因此表現都很接近，但隨著噪聲級別階段的降低，以及訓練的 epochs 數目隨之增加，Fast 也很明顯地與 Gradual 和 Slow 拉開距離。

此外，在 stage 2 也可以發現，Fast 在較晚的時候才暫停，而其餘兩者皆早早就開始收斂。

D. 最佳結果

從上述 B 和 C 我們可以得知在 Epochs 數有一定量的情況與 Epochs 數隨著噪聲級別增加的情況下的 Fast 噪聲級別設定會有相對較好的結果，因此我針對這兩者再做訓練比較。而在前面的訓練結果中，再 Epochs 有一定數量的情況比隨著噪聲級別增加的情況有較好的最終結果，這也十分合理，因為我設定隨著噪聲級別增加的情況他最一開始的初始訓練 epochs 數目相對較少，這可能會使得他在最一開始訓練成效相對的較差，儘管在後面的訓練成效差了一些，但總結來說還是較 Epochs 數有一定量的情況的訓練成果稍遜色，下為訓練成果比較。

(表七：Epochs 數量一定量與 Epochs 隨著噪聲級別增加兩種情況表現比較表)

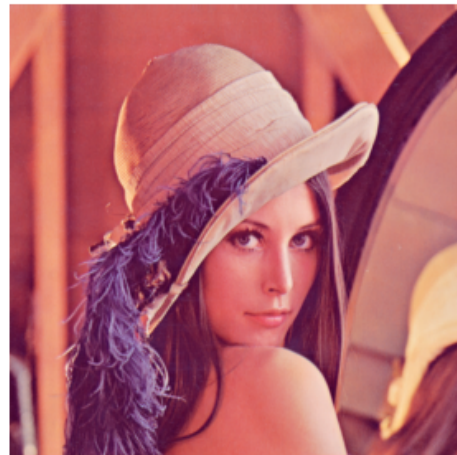
(Epochs 數量一定量其數量為每一個 stage 1000 epochs)

| | Gradual Epochs | Increasing Epochs |
|-----------------------------|----------------------------------------------------|----------------------------------------------------|
| PSNR Score | 41.8031 | 39.6282 |
| SSIM Score | 0.9816 | 0.9789 |
| Early Stopping state | At Epoch 690 in Stage 2 At Epoch 600 in Stage 3 | At Epoch 690 in Stage 2 At Epoch 620 in Stage 3 |

DIP Output



DIP Output



(圖十一 (左)：Epochs 數隨著噪聲級別增加在改良後 DIP 模型的最後圖片成果)

(圖十二 (右)：Epochs 數為一定量在改良後 DIP 模型的最後圖片成果)

圖片之成果由肉眼可以說相差無幾，只能從 PSNR 跟 SSIM 的分數來做判斷優劣。但是在與傳統 DIP 生產出的圖三來比較的話，皆可從肉眼上看出明顯的差距。

但根據時停的時間點來看，應該要是後者的分數來的高一些，這可能是因為在第一階段的時候，後者訓練的 Epochs 數量是 500，而前者則是 1000，才會導致這樣的情況發生。否則依照時停的時間點來看，若是我們隨著噪聲級別一路的增加我們 Epochs 數，應該會有最佳表現。

E. 實驗(實作)過程中遇到的挫折

在實作的過程中有時會遇到訓練失敗的問題，也就是最後 Renoise 後，模型輸出的結果圖片會是沒有任何圖案，純色顯示，而回去看他歷來訓練過程的 PSNR 分數跟 SSIM 分數就會發現它基本上只會有少許變動，但又不足以令我的判斷是去停止他，而發生這種情況常常在模型架構 layers 為 5 的時候。另外，我在實作 epochs 數目的比較時，若是使用的模型架構 layers 為 4 的時候，Fast 跟 Slow 有時也會訓練失敗，甚至結果跟我理論推測相差甚遠，直到我將模型架構的 layers 設定為 3 之後才逐漸穩定，這可能跟我在 A 提到的部分有關，可能是圖片相對較簡單，不需要太過複雜的模型架構訓練。

5. 結語

DIP 模型在有了類 DDPM 的監督信息後，可以明顯地看出無論是在無效的時間消耗上，抑或是最終圖片的成效上來看，皆相較於傳統 DIP 模型來說更佳的優良。經過改良後的模型，可以藉由噪聲級別加深加廣噪聲的訓練，也能藉此將傳統的 DIP 模型可能因為只能在相同的噪聲級別訓練，導致最終因為提前收斂，無法達到更好的訓練效果的問題。在改良後，每進入一個新的噪聲級別，我們就重新判斷一次是否需要 early stop，這使得整個訓練過程可以更加的完善，讓結果來的更好。總結來說，在本次實作將 DIP 與 DDPM 兩者結合帶來的成效無用置宜是 1 加 1 大於 2 的，將雙方各自的優點合併，DIP 的 early stopping method 跟受 DDPM inspired 的 Supervision，使得訓練的過程，更具效率與成效。

6. 參考文獻

- [論文導讀]Diffusion Model-Denoising Diffusion Probabilistic Models(DDPM)詳細介紹：<https://adam-study-note.medium.com/diffusion-model-denoising-diffusion-probabilistic-models-ddpm-詳細介紹-5ce77b6b64d4>
- 實作理解 Diffusion Model: 來自 DDPM 的簡化概念：<https://medium.com/ai-blog-tw/邊實作邊學習 diffusion-model-從 ddpm 的簡化概念理解-4c565a1c09c>
- [Day25]：擴散模型之 DDPM 原理介紹以及數學公式推導：<https://ithelp.ithome.com.tw/articles/10329715>
- DmitryUlyanov / deep-image-prior：<https://github.com/DmitryUlyanov/deep-image-prior>