在這份作業中，我的理解是DIP要把有噪音的圖片還原，所以在訓練時，我用原始圖片和DIP的輸出去算loss；DDPM是要去預測噪音，進而生成出沒有噪音的圖片。我選用的dataset是mnist dataset。

Theoretical Justification

我選用方法一，我把DIP model的output作為DDPM model的輸入。訓練的部分，把有雜訊的圖片給DIP還原，讓DDPM去預測雜訊，把預測的雜訊和0計算loss，因為要生成出沒有雜訊的圖片，但是我發現這方式產生的圖片其實也沒有很好，所以我設計了另一種計算loss的方式，把DDPM的預測雜訊拿去產生圖片，再把生成的圖片和沒有雜訊的圖片去計算loss，我覺得這樣就可以產生比較好的圖片。

Experimental Verification

Standalone DIP PSNR: 11.578571226350379, SSIM: 0.33443740589908816

Standalone DDPM PSNR: 16.09159592542095, SSIM: 0.7368279124152162

Integrate PSNR: 11.637577814345452, SSIM: 0.340805161893844

Integrate2 PSNR: 15.493682121224705, SSIM: 0.719116797426346

我的integrate是DIP和DDPM各自訓練完後，先用DIP修復有雜訊的圖片，在把修復的圖片丟給DDPM去生成；而另一個Integrate2是依照Theoretical Justification說的方法去訓練的，餵有雜訊的圖片給model，讓他去生成圖片。

Ablation Studies and Analysis

調整alpha\_t來看生成的圖片的效果

0.7  
Standalone DIP PSNR: 16.754250384336654, SSIM: 0.6428596466502621

Standalone DDPM PSNR: 22.615236886263656, SSIM: 0.8996644555668015

Integrate PSNR: 18.702268753360393, SSIM: 0.732189503136622

Integrate2 PSNR: 22.615236886263656, SSIM: 0.8996644555668015

0.5

Standalone DIP PSNR: 16.01464191527384, SSIM: 0.5634158111537214

Standalone DDPM PSNR: 6.7910256744639135, SSIM: 0.2783892261526309

Integrate PSNR: 15.964094895004411, SSIM: 0.5353257656328319

Integrate PSNR: 6.7910256744639135, SSIM: 0.2783892261526309

GitHub link: https://github.com/RayHsu1117/GAI\_Hw4.git

一張含有 螢幕擷取畫面, 正方形, 樣式, 像素 的圖片

自動產生的描述一張含有 螢幕擷取畫面, 樣式, 正方形, 黑與白 的圖片

自動產生的描述一張含有 螢幕擷取畫面, 符號, 正方形, 設計 的圖片

自動產生的描述一張含有 螢幕擷取畫面, 樣式, 正方形, 黑與白 的圖片

自動產生的描述上面這兩張是Integrate2產生的圖片與原始圖片跟有雜訊圖片的對比

一張含有 螢幕擷取畫面, 樣式, 黑與白, 正方形 的圖片

自動產生的描述一張含有 螢幕擷取畫面, 樣式, 正方形, 黑與白 的圖片

自動產生的描述

雜訊圖片與Integrate生成圖片的對比

雜訊圖片與DIP修復圖片的對比

雜訊圖片與DDPM生成圖片的對比