國立屏東大學
智慧機器人學系
Department of Intelligent Robotics

112學年度

專題研究期末報告

不同生成模型應用於圖像翻譯任務之比較與探討

組　員：何杰懋 (CBD109019)
謝昕諺 (CBD109023)
莊佩蓁 (CBD109031)
陳冠霖 (CBD109035)

指導老師：楊柏遠 博士

中 華 民 國 112 年 12 月

摘要

人工智慧與深度學習已然是目前科技發展的趨勢。在這之中，生成式 AI 的發展日新月異，從圖像生成、影片生成、聲音訊號生成、到聊天機器人等，都是各種不同的生成式 AI 所延伸出來的應用。生成式 AI 中又以圖像生成為最廣泛發展的領域。本專題研究分別應用不同的生成模型在圖像翻譯，並比較他們之間的效能差異；圖像翻譯是以一張圖片作為輸入，而後生成另一張不同風格的圖片。本研究使用生成對抗網路與擴散模型作為生成模型，並進一步地研究探討這些模型應用於圖像翻譯的能力以及優缺點。

關鍵字：生成式 AI、圖像翻譯、生成對抗網路、擴散模型

Abstract

Artificial intelligence (AI) and deep learning are currently the prevailing trends in technological development. Among these, generative AI has rapidly advanced, with various applications such as image generation, video generation, sound signal generation, and chatbots. Image generation is the most extensively developed field in generative AI. This project employs different generative models for image translation and compares their performance differences. Image translation involves taking an image as an input and generating another image with a distinct style. This research employs Generative Adversarial Network (GAN), Variational Auto-Encoder (VAE), and Diffusion Models as generative models. The study further investigates the abilities and advantages and disadvantages of these models as they are applied to image translation.

Keyword: generative AI, image translation, generative adversarial networks, diffusion model

目錄

圖目錄

表目錄

第1章　緒論

1.1　研究背景

生成式 AI 的發展速度飛快，且在人工智慧領域中非常具有潛力。雖然生成式 AI 是從2014年生成對抗網路被提出後才得到重視，但這幾年研究發展卻非常迅速。現階段各種 AI 生成的圖像充斥在生活周遭，另外因應人類不同任務需求，各種多樣化的生成模型被提出。也有基於各種理論而被研究出來的各種生成模型架構，例如使用編碼器與解碼器為主的變分自動編碼器、以兩個網路對抗訓練為主的生成對抗網路、基於非平衡熱力學中的擴散過程為基礎的擴散模型。這些模型的原理天差地遠，但都能生成出圖片來，對此，本研究旨在研究以這些模型為基礎的圖像生成模型，在圖像翻譯的任務中其模型的生成效能、生成品質差異等進行一個比較並分析不同模型間的優劣區間。

1.2　研究動機

圖像生成的模型在未來發展中勢必是一個重要的方向，本研究想對於不同種類的圖像生成模型進行研究，研究其理論以及建模方式，直到實際用於訓練資料的表現等進行完整的實驗，並從中學習到更深度的知識以及累積當今熱門研究領域之研究經驗。

1.3　研究目的

本研究希望透過比較不同的生成模型演算法去探討出對於圖像翻譯的生成任務，研究中會使用草圖資料輸入至模型中並生成真實圖片。接著探討哪些模型在生成上效果較優秀，哪些模型在圖像生成上能力較差。並透過完整的比較去分析出不同模型之間的架構、參數設定等造成的影響。

1.4　研究範圍與架構

本研究使用變分自動編碼器 (Variational Autoencoder, VAE)、條件式生

成對抗網路 (Conditional Generative Adversarial Networks, CGAN)、Pix2Pix、Cycle-GAN、去噪擴散隱式模型 (Denoising Diffusion Implicit Models, DDIM)模型去進行圖像翻譯的任務。資料集使用 façade labels→photo，這是一個將房屋外觀標籤化成一張草圖，以及對應的房屋外觀圖，整個資料集草圖與真實圖片各有400張圖片。

1.5 研究方法與流程

實驗中首先進行資料的預處理，因為資料只有400張圖片，故本研究對其做了資料增強，將資料量擴增為2400張。接著建立多種生成模型進行比較。最後使用不同指標對各生成模型所生成的圖片進行評估，並分析各模型的效能。

1.6 預期效益

本研究預計分析各種生成模型的模型效能，並從中挑出一個最適合用於圖像翻譯任務的模型，並統整所有模型在圖像翻譯任務中的優缺點。

第2章 文獻探討

自從深度學習的概念逐漸普及，人工智慧發展的速度也隨之提升許多。深度學習能夠達成的任務也非常廣泛，例如能夠使用深度學習來進行迴歸分析、分類預測、影像處理、資料生成、機器人控制[1]等。迴歸分析中，文獻[2]使用了多種迴歸的演算法來對現有的任務進行全面性的分析與概述。此外也有使用半監督式學習 (Semi-Supervised Learning, SSL)與交替學習(Alternate Learning)來解決深度學習模型的參數估計問題[3]，該研究所提出的方法能夠緩解樣本量不足的問題，使研究人員不需要再收集其他具有目標值的額外樣本。在分類預測中也有許多研究，例如文獻[4]使用深度卷積網路來根據肺部超音波的照片來判斷 COVID-19的分類與成像生物標記的定位等。

接著在影像處理中深度學習也能夠達成廣泛的任務，藉由卷積神經網路 (Convolutional Neural Network, CNN)[5]能夠更良好的處理圖片資料的特徵等訊息，也能將圖片透過深度卷積神經網路進行各種處理以萃取特徵等。

除了對圖片進行分析，深度學習也可以使用 CNN 將圖片進行編碼，編碼成一個固定長度的向量，接著再進行解碼，根據編碼內容透過 CNN 再反轉換回原始的圖片。這個應用即為自動編碼器 (Auto-Encoder, AE)，這個網路的概念分為編碼器與解碼器。編碼器的任務是將圖片編碼為一定長度的隱向量；解碼器的任務是將隱向量反轉變回原始圖片。這個做法有許多用意，可以降低資料傳遞時所佔用的空間，即將圖片透過網路傳給其他用戶時在伺服端會將資料壓縮成隱向量，接著傳送給另一用戶時再進行解碼將圖片反轉回原始圖片。另外也有許多應用，例如[6]使用基於注意力機制的 AE，該研究所提出的方法是用於預測藥物與疾病關聯的應用，方法中整合了藥物與疾病的關聯、不同疾病的相似度、不同藥物的相似程度、藥物的屬性等資料，根據這些資料建構了一個圖卷積自動編碼器網路，用於學習每個藥物與疾病的拓撲表示，藉此可以判斷藥物與疾病的關聯。雖然 AE 的用途廣泛，也可以生成資料。但是因為隱向量的分布狀況無法計算出來，人類無法模擬這些隱向量分布，從此分布中採樣一段編碼來讓解碼器生成資料。所以將 AE 變成生成模型還需要克服隱向量編碼的問題。

在2013年，變分自動編碼器 (Variational Auto-Encoder, VAE)[7]被提出來，這項研究增加了編碼器編碼的限制，使其編碼的結果需要符合常態分布。如此一來就可以從常態分佈中取樣一段編碼用於生成資料。自此之後自動編碼器能夠憑空生成各種圖片，是最基本的生成式 AI 應用。接著在2014年生成對抗網路 (Generative Adversarial Network, GAN)[8]被提出，這個生成模型架構使用兩個神經網路進行訓練，分別為生成器 (Generator)與判別器 (Discriminator)，生成器與判別器會相互對抗式的訓練，生成器訓練目標是根據常態分佈中採樣出來的雜訊來生成圖片，而判別器的訓練目標是判斷輸入圖片是來自資料集的真實圖片抑或是來自於生成器的假圖片。透過不斷的對抗訓練，生成器生成的圖片會越來越接近真實圖片；判別器的判斷能力也會越來越強大。訓練到最後達到納許均衡時，生成器與判別器的能力就不會再因訓練而進步，此時生成對抗網路模型即訓練完畢。在 GAN 被提出後過沒多久深度卷積生成對抗網路 (Deep Convolutional Generative Adversarial Networks, DCGAN)[9]也被提出，此模型使用了 CNN 來代替原始 GAN 使用全連接層的應用，使生成圖片的品質變得更好，目前

許多 GAN 的應用都是使用 CNN 為主架構。此外在 GAN 提出後的幾年中，關於 GAN 的變種與應用研究呈現指數型的增長，代表著 GAN 在生成式 AI 領域的潛力無窮。在能夠生成圖片後學術界開始致力於發展條件式的生成，使 GAN 的生成結果能夠被人類控制。在此之下條件式生成對抗網路 (Conditional Generative Adversarial Network, CGAN)[10]也在2014年被提出。此模型將條件向量與雜訊一同被輸入至生成器中，藉由條件遷入使生成結果變的可控；另外判別器也要訓練圖片的真假以及圖片是否有照條件生成。在 GAN 的發展大致確認後研究人員開始將不同架構、不同目標函數、不同預訓練模型等加入 GAN，希望以此能夠更符合的生成人類要求的圖片。在2016年，基於 CGAN 的模型 Pix2Pix 被提出[11]，與 CGAN 不同的是 Pix2Pix 直接將圖片輸入，藉此輸出不同風格的圖片。除此之外 Pix2Pix 還在生成器中使用 U-Net 架構使生成器能夠更細緻的處理圖片特徵的細節，此外在判別器中使用 PatchGAN 的架構讓判別器不再是對一張圖片判斷真假而是對部分圖片判斷真假，讓模型能夠知道生成圖片哪個部分還不夠逼真。在 Pix2Pix 被提出之後，2017年同一作者又提出了 Cycle-GAN[12]，這個模型改良了 Pix2Pix 只能使用成對數據來訓練的缺點，Cycle-GAN 使用兩組生成器與判別器分別訓練其風格變化的特徵，在目標函數上使用了迴圈一致性的概念使圖片在經過兩次風格轉換後能夠再次變回原圖。此外在模型架構上除了使用 U-Net 以外還使用了殘差的概念，使模型更深層卻又可以避免因網路層太深而導致梯度消失。透過這些概念 Cycle-GAN 能夠更良好的達成風格轉換的任務，生成的圖片品質又更加優秀，並且在資料集的準備上能夠更有彈性。在2020年，擴散模型被提出，其中最原始的去噪擴散機率模型 (Denoising Diffusion Probabilistic Models, DDPM)[13]是目前擴散模型的始祖，他透過擴散時間為圖片加上噪音，使圖片在經過一定擴散時間後完全變為一張雜訊圖片。接著訓練神經網路來模擬前向擴散中加上雜訊的平均期望分布，根據此分布生成逆向擴散的雜訊分布並應用於雜訊去噪，經過一定時間的去噪後雜訊圖片即會被去噪變回原始圖片。該項研究的生成能力甚至比 GAN 更好也更穩定，所以目前生成模型大多數都使用擴散模型來生成圖片。不過在模型擴散過程中的採樣時間造成模型訓練相當久、計算量非常龐大，於是有一些研究探討如何更高效的採樣、擴散。2020年

底去噪擴散隱式模型 (Denoising Diffusion Implicit Models, DDIM)[14]被提出。與 DDPM 不同，DDIM 將採樣過程不再限制於馬可夫鏈，使採樣速度提高。接著計算出逆向擴散的隱式解，使生成結果變的固定，DDIM 的改良對於加速訓練有著非常大的幫助。

本研究使用 CGAN、VAE、Pix2Pix、Cycle-GAN、DDIM 來進行圖片的風格變換任務，將圖片草圖經過模型計算後生成解析度較高的圖片，並使用多種訓練指標來評估生成結果。並研究目前模型在訓練資料集的表現與限制，透過全面的分析來探討模型的效能與優缺點。

第3章　研究方法

3.1　資料前處理

本研究使用 façade labels→photo [15]資料集作為訓練資料，這個資料集包含一張草圖與一張真實圖片，資料集的圖片如圖 1。

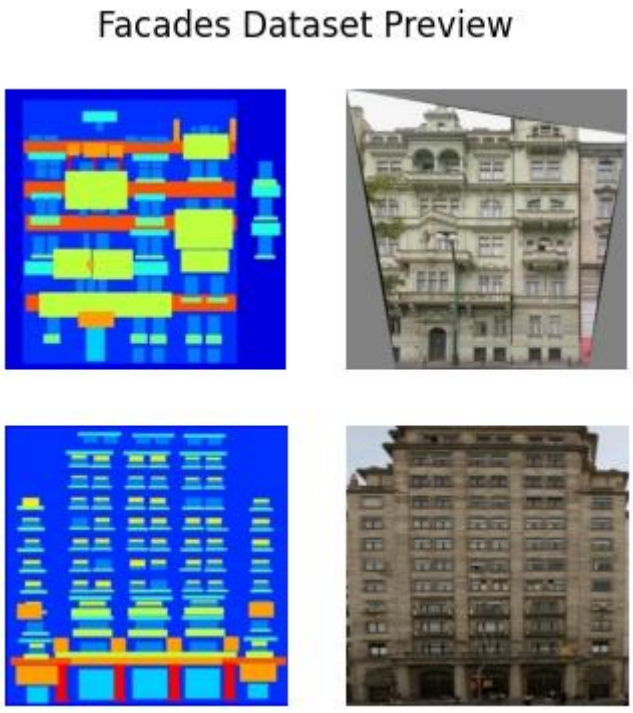


圖 1. 資料集內容，左邊為草圖標籤；右圖為真實圖片。

實驗中使用了資料增強，將圖片放大到300×300的解析度，再隨機從圖片中取256×256的區塊作為新圖片、以及將圖片進行鏡像水平翻轉。將資料擴充到2400張照片。

3.2 條件式變分自動編碼器 (CVAE)

3.2.1. 自動編碼器 (Auto Encoder, AE)

內文

3.2.2. 變分自動編碼器 (Variational Auto-Encoder, VAE)

內文

3.2.3. 條件式變分自動編碼器

內文

- 3.2.1、3.2.2前面簡單介紹 AE、VAE
- 3.2.3第一段介紹如何加入條件、用圖片輔助說明。描述模型架構，表格可以參考我的，我的表格字體只有9，再大會爆掉。
- 3.2.3第二段寫損失函數（重構跟 KL，KL 寫程式碼的設定方式(可以用 Pseudocode)，然後把完整推導過程寫在附錄)。方程式可以用多列表達，不要在一格中換行，可以複製我的方程式表格來用。
- 3.2.3第三段寫訓練的一些細節跟設定等。

3.3 條件式生成對抗網路 (CGAN)

CGAN 是一個 GAN 的種類，

- 第一段介紹 GAN，再來介紹 CGAN，描述模型架構，表格可以參考我的，我的表格字體只有9，再大會爆掉。
- 第二段寫損失函數，跟原始 GAN 幾乎一樣，只是變成條件機率的寫法，要具體說明優化目標為何。還有解釋所有參數。如果不會寫 latex 可以手寫拍照，但請拍清楚一點。
- CGAN 論文：https://arxiv.org/pdf/1411.1784.pdf。
- 第三段寫訓練的一些細節跟設定等。

$$\min_G \max_D V_{CGAN}(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z|y)\right)\right)\right] \quad (1)$$

3.4 Pix2Pix

Pix2Pix [11]是2016年被提出的生成模型，他是基於 CGAN 而優化改良的模型。該模型只接受一對一的資料被送入並進行圖像翻譯的任務。此模型分為兩個網路，生成器與判別器，其模型架構如表 1。架構之層數代號如

表 2所示，其中若沒特別說明卷積層與反卷積層的步進值，則步進值 m 為2；LeakyReLU 的負數部分斜率為0.2；每個卷積層與反卷積層的卷積核大小都為 (4,4)。生成器使用 U-Net 結構，此結構類似於 AE 網路，只是在下採樣層與上採樣對應層中會進行跳接，使梯度能夠傳達到前面的網路，並使下採樣的特徵與上採樣的圖片生成能夠有對應的關係。判別器使用 PatchGAN 架構，此架構會將輸入圖片分成許多部分，並判斷這些部分的真假，並相加以此判斷整張圖片的真假。不同於以往的 GAN 對於一張完整圖片判斷真假，此作法的好處是能夠更兼顧到細節的判斷。

Pix2Pix 中的生成器學習的目標函數如 (2)式所述，其中分為兩個部分，第一個是原始 CGAN 的目標函數，以及另一個 L1損失 (3)，該損失是生成器會學習將風格轉換後的圖片與真實圖片進行逐個像素的比較，比較生成圖片之平均絕對誤差，此即希望生成器能夠根據輸入圖片生成一張對應且正確的圖片。公式中 w 代表圖片寬、h 為圖片高、y_{ijk} 為真實圖片在 (i, j, k) 座標的像素值、y_{ijk} 會與生成圖片 $G(x)$在同樣座標下的像素值進行相減並取絕對值最後計算平均即為真實照片與生成照片之 L1損失。此損失會乘上一個超參數 λ，本實驗中 λ 值設定為100。

本研究的 Pix2Pix 模型訓練超參數設定如表 3，其中訓練80000次，每次訓練一批資料只使用1對圖片，優化器使用 Adam 優化器，生成器與判別器學習率皆為0.0002。

表 1. Pix2Pix 之網路架構

網路名稱	層數	網路層架構	輸出形狀
生成器 *G*	1	Generator input	(256, 256, 3)
	2	C64-L	(128, 128, 64)
	3	C128-IN-L	(64, 64, 128)
	4	C256-IN-L	(32, 32, 256)
	5	C512-IN-L	(16, 16, 512)
	6	C512-IN-L	(8, 8, 512)
	7	C512-IN-L	(4, 4, 512)
	8	C512-IN-L	(2, 2, 512)
	9	C512-IN-L	(1, 1, 512)
	10	C512-IN-L	(1, 1, 512)
	11	CT512S1-IN-D50-R, Concat 9	(1, 1, 1024)
	12	CT1024-IN-D50-R, Concat 8	(2, 2, 1536)
	13	CT1024-IN-D50-R, Concat 7	(4, 4, 1536)
	14	CT1024-IN-R, Concat 6	(8, 8, 1536)
	15	CT1024-IN-R, Concat 5	(16, 16, 1536)
	16	CT512-IN-R, Concat 4	(32, 32, 768)
	17	CT256-IN-R, Concat 3	(64, 64, 384)
	18	CT128-IN-R, Concat 2	(128, 128, 192)

	19	CT3-Tanh	(256, 256, 3)
判別器 ***D***	1-1	Sketch data input	(256, 256, 3)
	1-2	Discriminator input	(256, 256, 3)
	2	Concat 1-1 & 1-2	(256, 256, 6)
	3	C64-L	(128, 128, 64)
	4	C128-IN-L	(64, 64, 128)
	5	C256-IN-L	(32, 32, 256)
	6	Zero-Padding	(34, 34, 256)
	7	C512S1-IN-L	(31, 31, 512)
	8	Zero-Padding	(33, 33, 512)
	9	C1S1	(30, 30, 1)

表 2. Pix2Pix 網路架構代號對照

代號名稱	對應意思
CnSm	具有n個神經元的卷積層，步進值為m
L	LeakyReLU激活函數層
IN	實例正規化層
CTnSm	具有n個神經元的反卷積層，步進值為m
Dn	n%的丟棄層
R	ReLU激活函數層
Concat n	與第n層的輸出做合併，用於生成器
Concat n & m	將n與m層輸出做合併，用於判別器
Tanh	Tanh激活函數層
Zero-Padding	零填充層

表 3. Pix2Pix 之訓練超參數設定

參數名稱	參數值
訓練批次量	1
訓練次數	80000
生成器學習率	0.0002
生成器β_1	0.5
生成器β_2	0.999
判別器學習率	0.0002
判別器β_1	0.5
判別器β_2	0.999
L1損失權重λ	100

$$\mathrm{G}^{*}_{\mathrm{Pix2Pix}} = \arg\min_{G}\max_{D}\mathcal{L}_{CGAN}(G,D) + \lambda\mathcal{L}_{L1}(G) \tag{2}$$

$$\mathcal{L}_{\mathcal{L}1}(\mathrm{G}) = \frac{1}{3wh}\sum_{i=1}^{w}\sum_{j=1}^{h}\sum_{k=1}^{3}|y_{ijk} - G(x)_{ijk}| \tag{3}$$

3.5 Cycle-GAN

Cycle-GAN 也是一個 GAN 的變種

- 第一段介紹 Cycle-GAN，描述模型架構，印象中用了16層殘差?如果發現表格長度太長再想一下如何簡化，可以參考我的 DDIM，我的表格字體只有9，再大會爆掉。
- 第二段寫損失函數，所有損失都一樣要寫，可以分成多行但不可換行，不會寫 latex 可以手寫拍照，但請務必清晰，要具體說明優化目標為何。內文中不能有公式，所有公式都要用中文完整、詳細的解釋一遍。
- Cycle-GAN 論文：https://arxiv.org/pdf/1703.10593.pdf
- 第三段寫訓練的一些細節跟設定等

3.6 去噪擴散隱式模型 (Denoising Diffusion Implicit Models, DDIM)

DDIM [14]是近年來非常引人關注的生成模型，其基礎建立在擴散模型與去噪擴散機率模型上 (Denoising Diffusion Probabilistic Models, DDPM) [13]。其優秀的生成能力使目前研究都熱衷於探討這個模型，不過此模型的最大缺點就是訓練時間非常久、且使用的運算量相當龐大、也相當看重資料及的數量，若資料及不足則容易因為無法學習到特徵導致訓練失敗。

擴散模型的基本原理是利用前向擴散將一張照片添加雜訊，直到圖片變成幾乎成為雜訊；接著訓練神經網路預測此雜訊並使用逆向擴散將雜訊恢復成原始圖片。前向擴散的示意如圖 2，公式如 (4)。其中$q(\mathrm{x}_{1:\mathrm{T}}|\mathrm{x}_0)$為前向擴散中根據原圖 x_0到擴散時間1~T的條件分布形式，也代表原圖與各添加雜訊圖片的對應關係，添加雜訊的雜訊分布為一個常態分布 (6)，其平均為$\sqrt{1-\beta_t}$、變異數為β_t，β 代表擴散時間中的一個數列，通常值由0到1，也是添加雜訊的程度，β_t即為數列中對應擴散時間 t 的值。

圖 2. 對資料進行前向擴散之示意圖

接著為了學習到由一張雜訊的圖片返回成原圖的辦法，故需要建立一個深度學習模型來學習雜訊分布的狀況，並用於逆向擴散，逆向擴散的分布為 (6)，它也為一個常態分布 (7)，其中將雜訊圖片 x_t與擴散時間 t 輸入至

神經網路計算出$\mu_\theta(x_t,t)$並和$\sqrt{1-\beta_t}x_{t-1}$進行 KL 散度的相似度計算。至於前向擴散與逆向擴散的變異數經過論文作者實驗，結果非常相似，故不需要計算誤差。擴散模型最終的目標損失函數為 (8)，$\bar{\alpha}_t$是用於表達前向擴散到最後的雜訊 x_t 與原圖 x_0的計算參數；z 為前向擴散之雜訊，z_θ 為模型預測之雜訊。這些公式的詳細推導過程如附錄4。

DDIM 改良了 DDPM 在前向擴散中使用馬可夫鏈導致採樣時間很長的缺點。做法並非使用馬可夫鏈必須要知道前一個擴散時間的分布才能得知當下擴散時間的分布，如圖 3。它可以透過計算得知以初始條件 t_0的圖，直接計算出特定時間步的加雜訊圖片 。

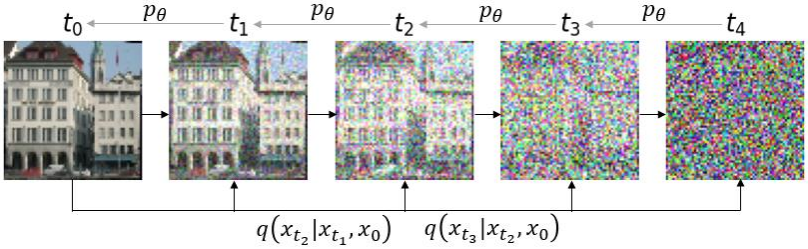


圖 3. DDIM 採樣之示意圖

本實驗之 DDIM 模型之擴散時間定義使用餘弦時間表，將擴散時間 t 變成0~1之間的連續值，並非 t=1,2,3…的離散值，首先定義餘弦角度之最大值$angle_{max}$與最小值$angle_{min}$，接著將當下的擴散時間透過 (9)計算為角度$angle_t$。接著計算這個角度的 sin 值與 cos 值，cos 值對應$\sqrt{1-\beta_t}$；sin 值對應$\sqrt{\beta_t}$。對應擴散時間 t 下的雜訊圖片則由 (10)計算，其中 *Image* 為原始資料集圖片；$\mathcal{N}$為從常態分佈採樣的雜訊。逆向擴散計算方式則使用 (11)計算，$\mathcal{N}_{\mathrm{p}\theta}$為神經網路預測之雜訊分布。經過一定次數去噪就能夠將 *Image* 還原成原始圖片。

本實驗 DDIM 之類神經網路架構如下表 4，模型主要使用了 U-Net 架構並搭配殘差區塊建立深層的模型。每個殘差區塊都有兩個卷積層，此外殘差的跳接部分也會再使用一個卷積層轉換。網路架構的代號對照如表 5，網路區塊的輸出形狀中 w, h, c 分別代表輸入資料的寬、高、通道數。上採樣區塊中上採樣層使用雙線性差值法。網路對於擴散時間輸入會經過正弦曲線嵌入層在使用上採樣層把輸出大小擴大成相應尺寸，其中上採樣的插值

方法使用最近鄰差值。另外正弦函數曲線嵌入層會透過將擴散時間 t 作為輸入，接著經過計算嵌入至長度為32的向量，計算方法如(12)-(13)所示。其中(12)代表一個初始值為0，終止值為3，公差 d 為0.2，共有16個項目的等差數列。方程式(13)是將這些數列中的所有元素 a_t 都計算出$2\pi e^{a_t}$的值，再分別計算 sin 與 cos 的值，並整理成變成一個長度為32的向量。

本實驗 DDIM 參數使用如下表 6 ，其中每次訓練的批次量為2，每個 epoch 會完整的訓練一次資料集，故只有50輪的訓練。優化器使用 AdamW，這是 Adam 的變種之一，其加入了權重衰減係數，值為0.00005。其他參數與 Adam 相同，學習率為0.0001；β_1為0.5；β_2為0.999。另外擴散時間使用餘弦擴散時間表，故會限制其角度的上下限，角度最大值為 $\cos^{-1}(0.95)$，也就是約0.31756 rad，角度最小值為 $\cos^{-1}(0.02)$ ，也就是約1.55079 rad。

表 4. DDIM 之網路架構

網路 / 區塊名稱	層數	網路層架構	輸出形狀
殘差區塊 R	1	Input	(w, h, c)
	2-1-1	BN	(w, h, c)
	2-1-2	Cnk3	(w, h, n)
	2-1-3	Swish	(w, h, n)
	2-1-4	Cnk3	(w, h, n)
	2-2	Cnk1	(w, h, n)
	3	Add 2-1-4 & 2-2	(w, h, n)
下採樣區塊 D	1	Input	(w, h, c)
	2	Rn	(w, h, n)
	3	Rn (Concat input)	(w, h, n)
	4	AveragePooling Size=2	(w/2, h/2, n)
上採樣區塊 U	1	input	(w, h, c)
	2	Up-Sampling Size=2 Interpolation used “bilinear”	(2w, 2h, c)
	3	Concat D_3	(2w, 2h, c+c_{D3})
	4	Rn	(2w, 2h, n)
	5	Rn	(2w, 2h, n)
DDIM 網路架構	1-1	Noise input	(64, 64, 3)
	1-1-2	C32k1	(64, 64, 32)
	1-2	Diffusion time input	(1, 1, 1)
	1-2-1	Sinusoidal embedding	(1, 1, 32)
	1-2-2	Up-Sampling Interpolation used “nearest”	(64, 64, 32)
	2	Concat 1-1-3 & 1-2-2	(64, 64, 64)
	3	D64	(32, 32, 64)
	4	D128	(16, 16, 128)
	5	D256	(8, 8, 256)
	6	D256	(4, 4, 256)
	7	R512	(4, 4, 512)
	8	R512	(4, 4, 512)
	9	U256 Concat 6	(8, 8, 256)
	10	U256 Concat 5	(16, 16, 256)
	11	U128 Concat 4	(32, 32, 128)
	12	U64 Concat 3	(64, 64, 3)
	13	C3k1	(64, 64, 3)

表 5. DDIM 網路架構代號對照

代號名稱	對應意思
Cnkm	具有n個神經元的卷積層，卷積核大小為(m, m)
Swich	swish激活函數層
BN	批次正規化層
Add n & m	將輸入n與m對應元素相加
Concat n & m	將n與m層輸出做合併，用於判別器
Sinusoidal embedding	正弦曲線嵌入層
Up-Sampling	上採樣層，將輸入寬高放大size倍
AveragePooling	平均池化層，將輸入寬高縮小size倍
Concat c_{D3}	上採樣與對應下採樣第3層的輸出做合併
Dn	下採樣區塊，其中每個卷積層都有n個神經元
Rn	殘差區塊，其中每個卷積層都有n個神經元
Un	上採樣區塊，其中每個卷積層都有n個神經元

表 6. DDIM 之訓練超參數設定

參數名稱	參數值
訓練批次量	2
訓練次數	50
學習率	0.0001
優化器β_1	0.5
優化器β_2	0.999
權重衰減係數	0.00005
擴散時間之$angle_{min}$	$\cos^{-1}(0.02)$
擴散時間之$angle_{max}$	$\cos^{-1}(0.95)$

$$q(\mathrm{x}_{1:\mathrm{T}}|\mathrm{x}_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{4}$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{5}$$

$$p(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{6}$$

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{7}$$

$$L_{simple}(\theta) = \mathbb{E}_{t,x_0,z}\left[\left\|z - z_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z, t\right)\right\|^2\right] \tag{8}$$

$$angle_t = angle_{min} + t(angle_{max} - angle_{min}) \tag{9}$$

$$\mathrm{N_t} = \sqrt{1-\beta_\mathrm{t}} \cdot \mathrm{Image} + \sqrt{\beta_\mathrm{t}} \cdot \mathcal{N} \tag{10}$$

$$\text{Image} = \frac{N_t - \sqrt{\beta_t} \cdot \mathcal{N}_{\text{p}\theta}}{\sqrt{1-\beta_t}} \tag{11}$$

$$a_t = \overbrace{\{0,0.2,0.4,\cdots 3\}}^{d=0.2,n=16} \tag{12}$$

$$e_t = \{\overbrace{\sin(2\pi e^{a_t}), \cos(2\pi e^{a_t})}^{n=32}\} \tag{13}$$

3.7 Adam 優化器

Adam 優化器[16]是深度學習中常用的優化器之一，也是本研究中使用的優化器。Adam 優化器結合了其他優化器例如 RMSProp 與 AdaGrad 的優點，其他優點包括可以自動調整學習率、可應用於不穩定的目標函數、相比其他優化器較不容易陷入局部最佳等。此優化器會對梯度的平均值與梯度的變異數進行考慮並計算下次更新的值。

Adam 優化器更新權重參數的方式如方程式(14)-(22)。方程式 (14)是第 t 訓練時間的梯度，其中 θ 是訓練的深度學習模型內的參數。根據這個梯度接著計算梯度的一階矩估計值 (First moment estimate) (15)，初始化的 m_0為0，其中 β_1是可設定的超參數，用於控制權重的分配。若 β_1值接近0則會較依賴當前的梯度，反之接近1則會較依賴於先前的權重。接著使用方程式 (16)計算梯度的二階矩估計值 (Second raw moment estimate)，初始化的 v_0也等於0，公式中的 β_2可以控制梯度平方的影響狀況，接近1會更依賴於之前訓練之梯度平方估計，會使訓練初期時能夠比較穩定。由於 m_0和 v_0為0所以會導致 m_t與 v_t偏向於0，所以需要對 m_t與 v_t進行偏差的修正，降低這些偏差對於訓練的影響，修正的方式分別為 (17)和 (18)。最後則是更新權重參數的步驟，這一步會使用 (19)來進行更新，其中 α_{lr}為學習率；ϵ為一個很小的值，為了避免$\hat{v}_t$為0造成分母為0而使更新變的無意義，通常預設值為10^{-8}。

$$g_t = \nabla_\theta f(\theta_{t-1}) \tag{14}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{15}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{16}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{17}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{18}$$

$$\theta_t = \theta_{t-1} - \frac{\alpha_{lr}\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} \tag{19}$$

3.8 模型評估指標

本研究使用多種不同的模型評估指標來評估生成模型生成圖片之圖片品質。

3.8.1 結構相似性指數

結構相似性指數（Structural Similarity Index, SSIM），為比較基礎的相似度指標，其顧名思義是用於計算兩張圖片之間其結構的相似性，此指標以圖片的亮度、對比度以及結構為計算的核心，透過考慮這三個因素來計算圖片失真的程度，這個指標因為考慮結構等因素，所以計算出來的結果會更符合人類的感知。

SSIM 的計算分為三個部分，分別為：亮度 (20)、對比度 (21)、結構 (22)。以這三個部分的結果為基礎再來計算 SSIM (23)。亮度計算中，μ_x 與 μ_y 分別為圖片 x 與圖片 y 的像素平均值，C_1為常數；對比度計算中，σ_x 與 σ_y 為圖片的標準差，C_2為常數；結構計算中，σ_{xy} 為兩張圖片的共變異數，C_3為常數。最後 SSIM 計算中 α、β、γ 都是超參數，可以決定指標中各部分的權重值。

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{20}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{21}$$

$$s(x,y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{22}$$

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{23}$$

3.8.2 峰值訊噪比

峰值訊噪比（Peak Signal-to-Noise Ratio, PSNR）是用來評估兩張圖片相似程度的指標。這個指標將圖片以訊號處理的方式計算其相似度，將圖片訊號以 PSNR 計算後的單位可視為分貝數，並依據分貝大小來判

斷圖片的相似度。通常結果為30dB~50dB 時圖片差異肉眼較難看出；低於30dB 時肉眼可以明顯看得出圖片的不同。

PSNR 的其公式為 (25)。其中 x 為生成圖片、y 為真實圖片。Max_y^2 是真實圖片中像素最大值的平方，乘以3是因為圖片為彩色，色彩通道為3。計算完成後會除以真實圖片與生成圖片的 L2誤差，也就是 MSE(x,y)，其計算方式如 (24)。其中 w 為圖片的寬度，h 為圖片的高度，接著將兩張圖片中每個像素的差異平方計算出來再計算算數平均，即為所求。

$$MSE(x,y) = \frac{1}{3wh}\sum_{i=1}^{w}\sum_{j=1}^{h}\sum_{k=1}^{3}\left(x_{i,j,k} - y_{i,j,k}\right)^2 \tag{24}$$

$$PSNR(x,y) = 10\log_{10}\left(\frac{3Max_y^2}{MSE(x,y)}\right) \tag{25}$$

3.8.3 Kernel Inception Distance (KID)

KID 是基於 Inception 網路的計算方式，Inception 網路是一個已經訓練好的深度學習模型，其使用 ImageNet 資料集[17]來做學習，資料集總計有1000種物件的分類，資料量共超過一百萬張圖片。但在計算 KID 時只會使用其中部分網路層以得到圖片的特徵圖。

KID 可以透過計算 Inception 網路出來的圖片特徵，將生成圖片與真實圖片特徵的平均值差異之平方計算出來並衡量兩個特徵之間的差異。此外 KID 還有一個三次核的無偏估計值，這個估計值能夠讓計算出來的結果更貼近人類的感知。所以 KID 總結來說即為將 Inception 網路計算出來之特徵向量空間的多項式核函數平方的最大平均差異 (Maximum Mean Discrepancy, MMD) (26)。其中 P 與 Q 分別代表生成圖像分布與真實圖像分布；H 為一個希爾伯特空間。方程式的目的為希望找到一個函數 f，其中 f 期望兩個機率分布$E_p f(x)$與$E_Q f(y)$的動差相減可以達到最大值，即求得期望之最小上界 (sup)。其中 f 屬於希爾伯特空間 H，且 f 限制在再生希爾伯特空間，其單位球$|f|_H \leq 1$。接著使用里斯表示定理 (Riesz's Representation theorem)將期望$E_p f(x)$改寫成希爾伯特空間的內積形式，如 (27)。其中 μ_p 為平均嵌入 (Mean Embedding)[18]，此時

MMD 可視為計算兩個分布在某個空間中 Mean Embedding 的距離。為了使這個距離最大，可以利用在希爾伯特空間中範數就是內積的原理來將原本的內積計算變成範數。然後在再生希爾伯特空間中內積可以使用核 (kernel)計算，最後整個公式可以變成平方 MMD 的表達式 (31)，也就是 KID 的計算方式。其中 m 為生成圖片數量；n 為真實圖片數量；$k(a,b)$為核 (kernel)的公式 (28)；d 為特徵向量維度，本實驗中使用 2048。

$$MMD(P,Q,H) = sup_{f\in H,|f|_H\leq 1} E_p f(x) - E_Q f(y) \tag{26}$$

$$E_p f(x) \leq f, H < E\big(\emptyset(x)\big) \leq f, \mu_p > H \tag{27}$$

$$k(a,b) = \left(\frac{1}{d}a^T b + 1\right)^3 \tag{28}$$

$$f_m(x) = \frac{1}{m(m-1)}\sum_{i\neq j}^{m} k(x_i, x_j) \tag{29}$$

$$f_n(y) = \frac{1}{n(n-1)}\sum_{i\neq j}^{n} k(y_i, y_j) \tag{30}$$

$$MMD^2(x,y) = f_m(x) + f_n(y) + \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i, y_j) \tag{31}$$

3.8.4 Fréchet Inception Distance (FID)

FID 分數是非常廣泛的圖像生成模型判斷的指標之一，其基礎也使用了 Inception 網路來萃取圖片的特徵。FID 適合評分生成模型的多樣性，但因為模型基於特徵提取所以並不會考慮特徵位置的合理性。比起 KID，FID 計算量較少，所以還是比較廣泛用於生成模型的訓練，FID 分數越低代表模型生成圖片之質量較佳。

FID 分數計算之公式為 (32)，其中$\hat{x}$與$\hat{y}$是生成圖片與真實圖片經過 Inception 網路計算出來的圖片特徵；μ 為平均值，公式會計算平均值差距的平方；接著會計算特徵的共變異數 σ，經過計算過後出來的結果為方陣，需要再計算其跡數 (Trace)。最後將總結果相加即為 FID 分數。

$$FID(x,y) = \left|\mu_{\hat{x}} - \mu_{\hat{y}}\right|^2 + Tr\left(\sigma_{\hat{x}} + \sigma_{\hat{y}} - 2\sqrt{\sigma_{\hat{x}}\sigma_{\hat{y}}}\right) \tag{32}$$

3.8.5 Learned Perceptual Image Patch Similarity (LPIPS)

Learned perceptual image patch similarity (LPIPS)[19]為計算兩張圖片的感知損失，這個指標也是基於深度學習模型提取特徵後的特徵相似度。比起不使用深度學習的 PSNR 與 SSIM，LPIPS 也能更貼近人類的感知。LPIPS 計算出來的值越低代表圖片的相似程度越高。

根據其原始論文，LPIPS 的公式為 (33)。其中 l 為從神經網路的第 l 層提取圖片特徵，H_l跟跟 W_l分別是該層出來的圖片特徵高與寬，作者使用此方法在通道維度上進行單位正規化。公式中使用向量 w 對逐個通道中經過激活函數計算的結果進行縮放，並計算 L2距離。若縮放的參數 w_l為1則代表計算餘弦距離。

$$LPIPS(x,y)=\sum_{l}\frac{1}{H_lW_l}\sum_{h,w}\left\|w_l\odot\left(\hat{x}^l_{h,w}-\hat{y}^l_{h,w}\right)\right\|_2^2 \tag{33}$$

第4章　研究成果

4.1　實驗結果

本研究

4.1.1　Pix2Pix 生成結果探討

本研究之 Pix2Pix 生成訓練使用兩種不同的損失進行訓練，並得到了不同的結果。第一種是與原始論文相同的設定，即 Pix2Pix 的鑑別器損失函數使用二元交叉熵損失 (BCE Loss)；生成器的對抗損失使用 L2，對於圖片的判斷依樣使用 L1損失。第二種是多方嘗試後修改的設定，即 Pix2Pix 的鑑別器損失函數使用 L2損失；生成器的對抗損失使用 L2，對於圖片的判斷依樣使用 L1損失。綜觀來說原始論文設定時訓練出來的圖片對於訓練資料集的生成會有肉眼可見的瑕疵；對於測試資料集的生成則較模糊。但使用修改過後的設定則對於訓練訓練集的擬合狀態較佳；對於測試資料集的生成則有一些瑕疵。對於測試資料的生成結果如圖 5。

研究中所修改的方法對於訓練資料的生成如圖 4所示，可以看到在

訓練中約20000次時生成器就幾乎學會生成訓練資料的分布了，但也因為這個高度擬合性所以在對於測試資料這些沒訓練過的圖片生成則會接近於訓練資料集，也就是說生成器可能過度擬合而造成無法生成具有多樣性的結果。

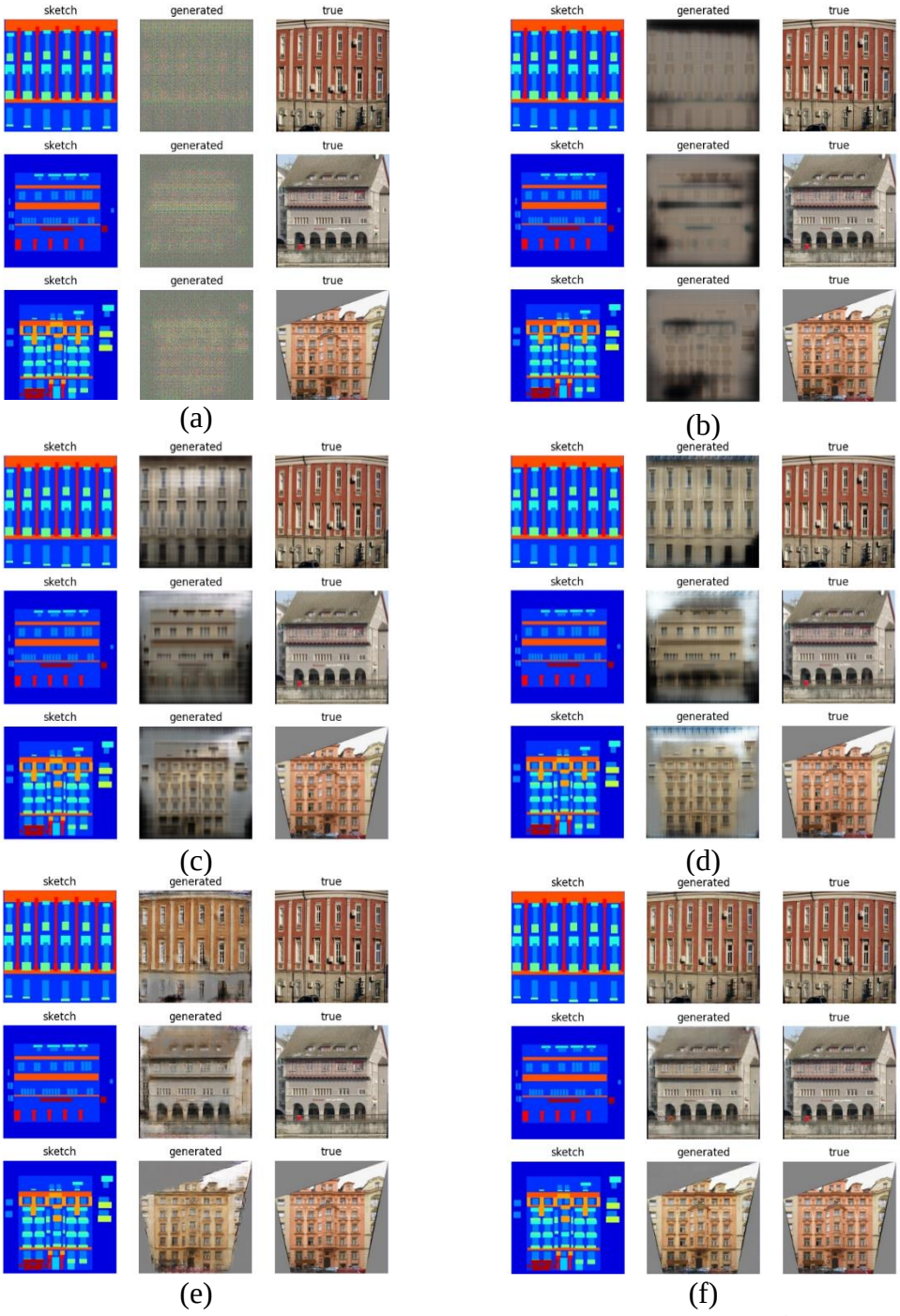
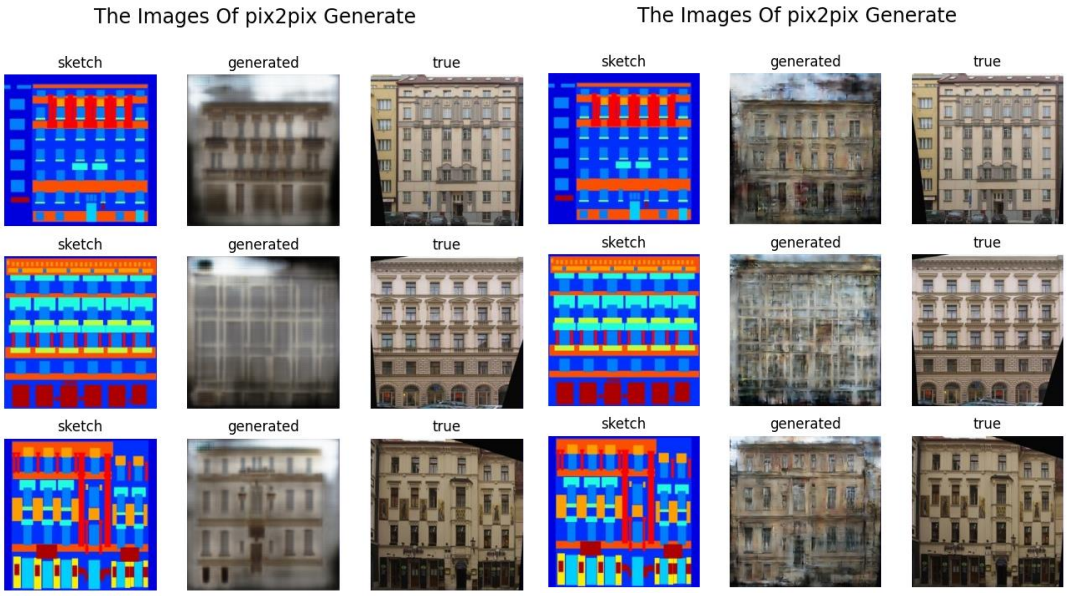


圖 4. 本研究修改之 Pix2Pix 對於訓練資料之訓練情形, (a) 第1次訓練, (b) 第200次訓練, (c) 第1000次訓練, (d) 第2500次訓練, (e) 第10000次訓練, (f) 第

20000次訓練。

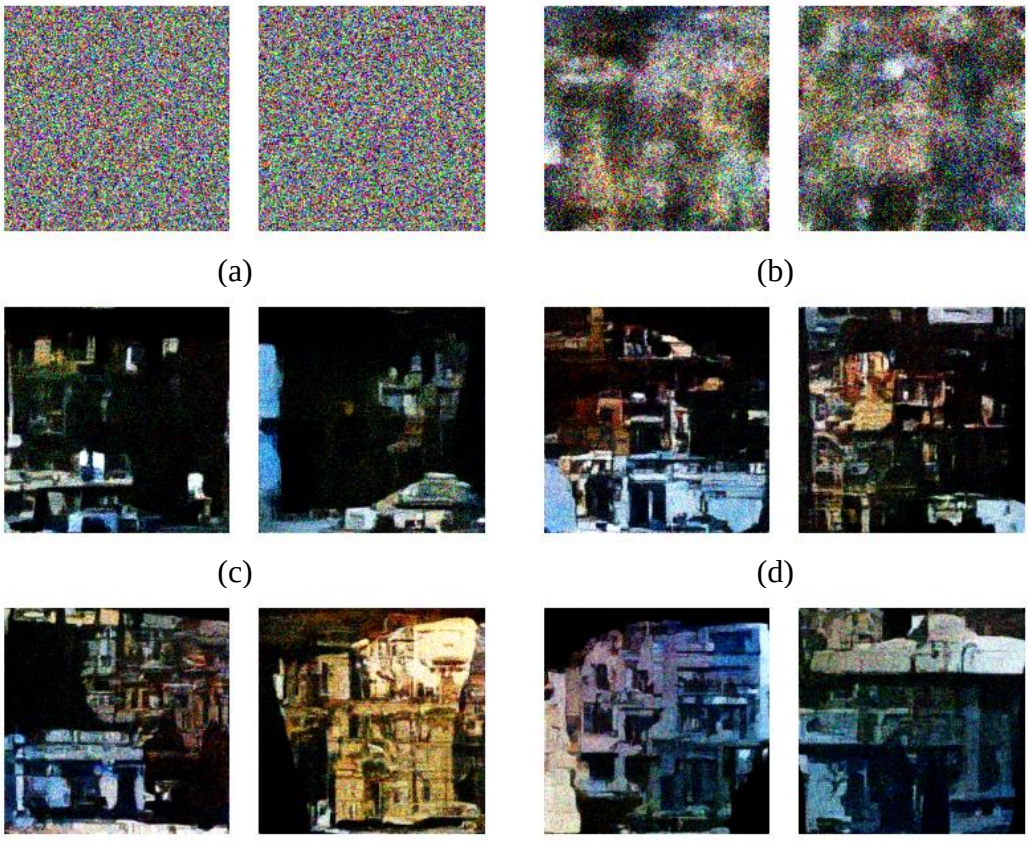


(a) (b)

圖 5. Pix2Pix 對於測試資料之生成能力比較, (a) 生成器損失使用 BCE+L1, (b) 生成器損失使用 L2+L1。

4.1.2 DDIM 生成結果探討

本研究同樣使用擴散模型來進行訓練生成，但目前生成效果不佳，推測可能原因為訓練資料量太少。使用擴散模型訓練生成圖片時對於資料集的數量有一定的要求，先前使用擴散模型訓練別的資料集時資料及數量都為50000筆到60000筆，而本研究之訓練資料集僅有400筆，就算使用資料擴增之後的2400筆資料似乎對於訓練的效果都不太好。另外訓練中發現使用解析度256×256的訓練資料對於運算的負擔太大，故未來會將資料解析度重整為64×64再進行訓練。目前訓練成果在一般無條件生成時就已經無法生成出良好的結果，訓練過程時所生成的結果如圖 6 所示。未來會繼續探討模型架構對於訓練生成造成的影響並再加入條件控制。



(a) (b)

(c) (d)

(e) (f)

圖 6. 本研究 DDIM 對於訓練資料之訓練情形, (a) 第1次訓練, (b) 第5次訓練, (c) 第15次訓練, (d) 第25次訓練, (e) 第35次訓練, (f) 第50次訓練。

目前訓練之損失函數變化如圖 7所示，其中 n_loss 曲線代表對於模型生成的雜訊損失、i_loss 曲線為圖片生成之像素誤差、kid 曲線是對於驗證資料集的 KID 分數計算，訓練時模型只根據 n_loss 進行梯度計算與反向傳播。由圖可知 KID 分數呈現震盪不穩的狀態，但其他損失卻有收斂，現階段本研究認為最於雜訊的生成模型可能已經學習到對應雜訊，但是在逆向擴散中因為沒有良好的學習到圖片的特徵等結果，所以在逆向擴散生成圖片時才會無法良好的生成正確的圖片。未來本研究會再嘗試透過降低解析度以換取更多一次訓練的批次量、使用資料增強生成更多資料等方式來實驗並分析結果。

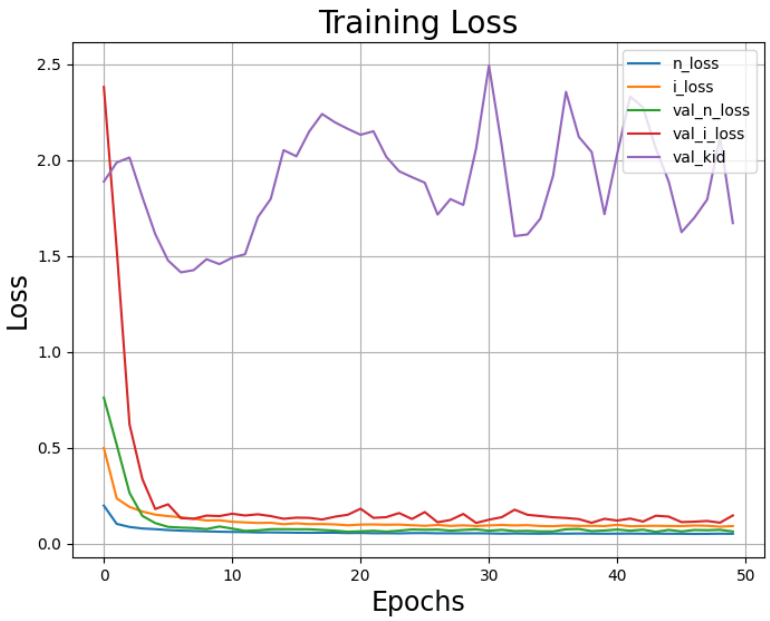


圖 7. 本研究 DDIM 之訓練損失變化。

第5章　結論與後續研究建議

5.1　結論

本專題使用多種生成模型來執行圖像轉圖像的圖像翻譯、風格變換類型任務，研究成果顯示…

在深層模型例如 CycleGAN 與需要大量運算資源的 DDIM 的訓練中，我們發現 CycleGAN 與 DDIM 訓練時間比起其他模型來說更久。而且 DDIM 訓練上也很容易因為資料量不夠而無法良好的學習到圖片的特徵，即使使用資料增強對於訓練時資料量可能還略顯不足。

5.2　後續研究建議

因為此資料集的資料量不算太多，所以對於一些深層網路的模型，或者需要大量資料集以用於學習圖片細節特徵的模型在訓練上會有一些特徵無法學習到。例如 DDIM 在訓練時效果就不太好，但在訓練有50000筆以上資料集的任務時效果就不錯。故我們推測可能是訓練資料量不足以訓練擴散模型，且硬體設備效能不足，所以訓練 DDIM 時需要花費更多時間。未來可以使用具有更多資料的資料集、並且將電腦設備提升以用於良好的訓

練 DDIM 類型的擴散模型。

參考文獻

[1] P.-H.Kuo andK.-L.Chen, “Two-stage fuzzy object grasping controller for a humanoid robot with proximal policy optimization,” *Eng. Appl. Artif. Intell.*, vol. 125, p. 106694, Oct.2023, doi: 10.1016/j.engappai.2023.106694.

[2] Y.Tai, “A Survey Of Regression Algorithms And Connections With Deep Learning,” Apr.2021, [Online]. Available: http://arxiv.org/abs/2104.12647

[3] A.Zell, G.Sumbul, andB.Demir, “Deep Metric Learning-Based Semi-Supervised Regression with Alternate Learning,” in *2022 IEEE International Conference on Image Processing (ICIP)*, IEEE, Oct. 2022, pp. 2411–2415. doi: 10.1109/ICIP46576.2022.9897939.

[4] S.Roy *et al.*, “Deep Learning for Classification and Localization of COVID-19 Markers in Point-of-Care Lung Ultrasound,” *IEEE Trans. Med. Imaging*, vol. 39, no. 8, pp. 2676–2687, Aug.2020, doi: 10.1109/TMI.2020.2994459.

[5] K.O’Shea andR.Nash, “An Introduction to Convolutional Neural Networks,” Nov.2015, [Online]. Available: http://arxiv.org/abs/1511.08458

[6] P.Xuan, L.Gao, N.Sheng, T.Zhang, andT.Nakaguchi, “Graph Convolutional Autoencoder and Fully-Connected Autoencoder with Attention Mechanism Based Method for Predicting Drug-Disease Associations,” *IEEE J. Biomed. Heal. Informatics*, vol. 25, no. 5, pp. 1793–1804, May2021, doi: 10.1109/JBHI.2020.3039502.

[7] D. P.Kingma andM.Welling, “Auto-Encoding Variational Bayes,” Dec.2013, [Online]. Available: http://arxiv.org/abs/1312.6114

[8] I. J.Goodfellow *et al.*, “Generative Adversarial Networks,” Jun.2014, [Online]. Available: http://arxiv.org/abs/1406.2661

[9] A.Radford, L.Metz, andS.Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” Nov.2015, [Online]. Available: http://arxiv.org/abs/1511.06434

[10] M.Mirza andS.Osindero, “Conditional Generative Adversarial Nets,” Nov.2014, [Online]. Available: http://arxiv.org/abs/1411.1784

[11] P.Isola, J.-Y.Zhu, T.Zhou, andA. A.Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” Nov.2016, [Online]. Available: http://arxiv.org/abs/1611.07004

[12] J.-Y.Zhu, T.Park, P.Isola, andA. A.Efros, “Unpaired Image-to-Image

Translation using Cycle-Consistent Adversarial Networks," Mar.2017, [Online]. Available: http://arxiv.org/abs/1703.10593

[13] J.Ho, A.Jain, andP.Abbeel, "Denoising Diffusion Probabilistic Models," Jun.2020, [Online]. Available: http://arxiv.org/abs/2006.11239

[14] J.Song, C.Meng, andS.Ermon, "Denoising Diffusion Implicit Models," Oct.2020, [Online]. Available: http://arxiv.org/abs/2010.02502

[15] "Facades Dataset," 2023. https://www.kaggle.com/datasets/balraj98/facades-dataset (accessed Sep.25, 2023).

[16] D. P.Kingma andJ.Ba, "Adam: A Method for Stochastic Optimization," Dec.2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[17] "ImageNet," 2023. https://www.image-net.org/ (accessed Sep.23, 2023).

[18] K.Muandet, K.Fukumizu, B.Sriperumbudur, andB.Schölkopf, "Kernel Mean Embedding of Distributions: A Review and Beyond," *Found. Trends® Mach. Learn.*, vol. 10, no. 1–2, pp. 1–141, 2017, doi: 10.1561/2200000060.

[19] R.Zhang, P.Isola, A. A.Efros, E.Shechtman, andO.Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," Jan.2018, [Online]. Available: http://arxiv.org/abs/1801.03924

[20] "Variational Bayesian methods," 2023. https://en.wikipedia.org/wiki/Variational_Bayesian_methods (accessed Sep.27, 2023).

誌謝

感謝

附錄

附錄1. 硬體設備

本研究使用之硬體設備如下表 7。

表 7. 研究使用之硬體設備資訊

設備名稱	設備資訊
CPU	intel (R) core (TM) i7-12700F
記憶體	32GB
GPU	NVIDIA GeForce RTX 3080 10GB

附錄2. 軟體環境

本研究中所使用之環境為 Python 3.8版本，整合開發環境 (Integrated Development Environment, IDE)為 Pycharm Professional 2021.1.2。研究中所使用到的套件版本如下表 8。

表 8. 研究使用之套件詳細資訊

套件名稱	版本編號
Numpy	1.19.2
Pandas	1.4.2
OpenCV	3.4.8.29
Keras	2.7.0
TensorFlow	2.7.0
TensorFlow-gpu	2.7.0
TensorFlow-addons	0.17.1
Matplotlib	3.5.2
protobuf	3.20.3

附錄3. 變分自編碼器 (VAE)之原理推導

附錄3-1. VAE 訓練之 KL 散度計算

VAE 訓練之 KL 散度計算為了要使程式編輯較好寫所以將傳統 KL 散度的公式展開改寫。以便因應利用編碼器模型的輸出與平均為0、變異數為1的常態分布進行 KL 散度的計算。KL 散度的公式如 (34)，其中 $p(x_n)$與 $q(x_n)$為兩個機率分布。

$$D_{KL}(p||q) = \sum p(x_n) log \frac{p(x_n)}{q(x_n)} \tag{34}$$

VAE 要計算編碼器輸出與$\mathcal{N}(0,1)$的距離。

附錄3-2. VAE 訓練之目標函數推導

VAE 訓練目標之完整推導過程較複雜，主要是使用變分下界 (Variational Lower Bound, VLB)，也稱 ELBO。優化負對數似然 (Negative Log Likelihood, NLL)。也是將 KL 散度給改寫，首先定義兩個條件分布的 KL 散度。

$$D_{KL}(q_\phi(z|x)||p_\theta(z|x)) = \int q_\phi(z|x) log \frac{q_\phi(z|x)}{p_\theta(z|x)} dz \tag{35}$$

於是可以將$p_\theta(z|x)$使用條件機率 (36)替換並改寫成 (37)。

$$p_\theta(z|x) = \frac{p(z,x)}{p(x)} \tag{36}$$

$$D_{KL}(q_\phi(z|x)||p_\theta(z|x)) = \int q_\phi(z|x) log \frac{q_\phi(z|x)}{p_\theta(z,x)} p(x)\, dz \tag{37}$$

接著利用對數率將真數裡面相乘的部分拉到對數函數外變成相加。

$$\int q_\phi(z|x) \left(log \frac{q_\phi(z|x)}{p_\theta(z,x)} + log p_\theta(x) \right) dz \tag{38}$$

然後將括號展開，接著因為$\int q_\phi(z|x) log p(x) dz = log p(x)$，所以方程可以變成：

$$log p_\theta(x) + \int q_\phi(z|x) log \frac{q_\phi(z|x)}{p_\theta(z,x)} dz \tag{39}$$

再來使用貝式定理 (40)改寫方程。

$$p_\theta(z,x) = p(x|z)p(z) \tag{40}$$

$$log p_\theta(x) + \int q_\phi(z|x) log \frac{q_\phi(z|x)}{p_\theta(x|z)p_\theta(z)} dz \tag{41}$$

使用對數率將分母的項次獨立出來，並且改寫成期望值之形式。

$$log p_\theta(x) + \mathbb{E}_{z\sim q_\phi(z|x)} \left[log \frac{q_\phi(z|x)}{p_\theta(x)} - log p_\theta(x|z) \right] \tag{42}$$

將期望值中第一項改寫成 KL 散度之形式，於是可以將原式改寫成

$$D_{KL}(q_\phi(z|x)||p_\theta(z|x)) = log p_\theta(x) + D_{KL}\big(q_\phi(z|x)\|p_\theta(z)\big) - \mathbb{E}_{z\sim q_\phi(z|x)}[log p_\theta(x|z)] \tag{43}$$

接著移項。

$$\mathbb{E}_{z\sim q_\phi(z|x)}[log p_\theta(x|z)] - D_{KL}\big(q_\phi(z|x)\|p_\theta(z)\big) = log p_\theta(x) - D_{KL}(q_\phi(z|x)||p_\theta(z|x)) \tag{44}$$

上述方程式中，對於訓練的目標，需要最大化生成資料分布的對數似然 $(log p_\theta(x))$，而且還要最小化真實資料與生成之分布的差異。所以損失可以定義成下式：

$$L(\theta,\phi) = -log p_\theta(x) + D_{KL}(q_\phi(z|x)||p_\theta(z|x)) \tag{45}$$

在變分貝葉斯方法 (Variational Bayesian methods)[20]中，這種損失函數被稱為變分下界或者證據下界 (evidence lower bound, ELBO)，方程式的 KL 散度會大於0，所以-L 是下界$log p_\theta(x)$。

$$-L = log p_\theta(x) - D_{KL}(q_\phi(z|x)\|p_\theta(z|x)) \le log p_\theta(x) \tag{46}$$

附錄4. 擴散模型之原理推導

附錄4-1. 推導前向擴散之原始圖片與雜訊圖片之關係

前向擴散時有公式 (4)和 (5)，由此可以得知圖片 x_t 的計算方式如下：

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}z_t \tag{47}$$

接著論文作者定義了$\alpha_t = 1-\beta_t$用於代替 β_t，所以原式變為：

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}z_{t-1} \tag{48}$$

$$x_{t-1} = \sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}z_{t-2} \tag{49}$$

將方程式 (49)帶入 (48)，可以得到方程式：

$$x_t = \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}z_{t-2} \tag{50}$$

所以根據此規律可以算出 x_t 與 x_0的關係，我們將$\alpha_t\alpha_{t-1}\dots\alpha_0$表達如下：

$$\bar{\alpha}_t = \prod_{s=1}^{t} a_s \tag{51}$$

將此方程帶入則 (50)會變為 (52)，所以可進一步表達成常態分布的形式 (53)。

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z \tag{52}$$

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, 1-\bar{\alpha}_t I\right) \tag{53}$$

接著移項計算 x_0與 x_t的關係，可以得知：

$$x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}z}{\sqrt{\bar{\alpha}_t}} \tag{54}$$

附錄4-2. 推導目標函數

目標函數與 VAE 類似，也是使用 ELBO 來優化 NLL，優化計算結束後會得到 (55)，計算過程如附錄3-2。

$$-logp_\theta(x_0) + D_{KL}(q(x_{1:T}|x_0)\|p_\theta(x_{1:T}|x_0)) \geq -logp_\theta(x_0) \tag{55}$$

再來將神經網路訓練出來的分布$p_\theta(x_{1:T}|x_0)$使用條件機率的公式計算成$\frac{p_\theta(x_{0:T})}{p_\theta(x_0)}$接著再將 KL 散度 (34)的公式展開。於是得到：

$$-logp_\theta(x_0) \leq -logp_\theta(x_0) + \mathbb{E}_q[log\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + logp_\theta(x_0)] \tag{56}$$

接著$logp_\theta(x_0)$與$-logp_\theta(x_0)$抵消。

$$\mathbb{E}_q[-logp_\theta(x_0)] \leq \mathbb{E}_q\left[log\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right] \tag{57}$$

再將不等號右邊的項目展開，於是得到：

$$\mathbb{E}_q\left[log\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right] = \mathbb{E}_q\left[log\frac{\prod_{t=1}^{T}q(x_t|x_{t-1})}{p_\theta(x_T)\prod_{t=1}^{T}p_\theta(x_{t-1}|x_t)}\right] \tag{58}$$

把將分母的$p_\theta(x_T)$移至外面：

$$\mathbb{E}_q\left[-logp_\theta(x_T) + log\frac{\prod_{t=1}^{T}q(x_t|x_{t-1})}{\prod_{t=1}^{T}p_\theta(x_{t-1}|x_t)}\right] \tag{59}$$

根據對數率，指數函數內的真數相乘可以視為不同指數函數函數相加：

$$\mathbb{E}_q\left[-logp_\theta(x_T) + \sum_{t=1}^{T}log\frac{q(x_t|x_{t-1})}{p_\theta(x_{t-1}|x_t)}\right] \tag{60}$$

於是可以將擴散時間 t=1的部分另外獨立。

$$\mathbb{E}_q\left[-logp_\theta(x_T) + \sum_{t>1}^{T}log\frac{q(x_t|x_{t-1})}{p_\theta(x_{t-1}|x_t)} + log\frac{(x_1|x_0)}{p_\theta(x_0|x_1)}\right] \tag{61}$$

然後我們將連加符號內的前向擴散之分布加入條件 x_0並使用貝式定理將結果展開：

$$\mathbb{E}_q\left[-logp_\theta(x_T) + \sum_{t>1}^{T}log\frac{q(x_{t-1}|x_t,x_0)}{p_\theta(x_{t-1}|x_t)}\frac{q(x_t|x_0)}{q(x_{t-1}|x_0)} + log\frac{(x_1|x_0)}{p_\theta(x_0|x_1)}\right] \tag{62}$$

接著也把$\frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}$獨立出來。

$$\mathbb{E}_q\left[-logp_\theta(x_T) + \sum_{t>1}log\frac{q(x_{t-1}|x_t,x_0)}{p_\theta(x_{t-1}|x_t)} + \sum_{t>1}\frac{q(x_t|x_0)}{q(x_{t-1}|x_0)} + log\frac{(x_1|x_0)}{p_\theta(x_0|x_1)}\right] \tag{63}$$

我們將第四項中的結果進行馬可夫鏈的計算，算出成果後如(64)式

$$\sum_{t>1}\frac{q(x_t|x_0)}{q(x_{t-1}|x_0)} = log\frac{q(x_T|x_0)}{q(x_1|x_0)} \tag{64}$$

根據對數率，指數函數內的真數相除可以視為不同指數函數相減，然後再消除重複的項目後會變成：

$$\mathbb{E}_q = \left[logq(x_T|x_0) - logp_\theta(x_T) + \sum_{t>1}log\frac{q(x_{t-1}|x_t,x_0)}{p_\theta(x_{t-1}|x_t)} - logp_\theta(x_0|x_1)\right] \tag{65}$$

於是再使用對數率將第一項與第二項合併，之後可以化成 KL 散度的形式，於是目標函數變成 (66)，這也是原始論文中提出的公式。不

過因為 L_T 無訓練參數，所以計算出來是常數，可以忽略；L_0為重構項目，計算出的值非常小，可以忽略。故接著要將 L_{t-1}重參數化，求得最終損失。

$$\mathbb{E}_q[L_T + L_{t-1} + L_0] \tag{66}$$

$$L_T = D_{KL}(q(x_T|x_0)||p_\theta(x_T)) \tag{67}$$

$$L_{t-1} = \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) \tag{68}$$

$$L_0 = -logp_\theta(x_0|x_1) \tag{69}$$

附錄4-3. 推導$q(x_{t-1}|x_t, x_0)$之分布平均與變異數

要求得$q(x_{t-1}|x_t, x_0)$的後驗機率分布，我們根據論文原文假設 (70)。

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I) \tag{70}$$

根據貝式定理，$q(x_{t-1}|x_t, x_0)$正比於下式。

$$\exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{\left(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0\right)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}\right)\right) \tag{71}$$

然後將方程內的項目都展開，並結合在一起後，我們將方程使用分配率整理成與 x_{t-1}相關的方程式，因為 x_t與 x_0並不會用到故整理成一個函數 C 用於表示。

$$\exp\left(-\frac{1}{2}\left[\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)x_{t-1} + C(x_t, x_0)\right]\right) \tag{72}$$

根據機率密度函數可以知道變異數$\tilde{\beta}_t$等於x_{t-1}^2項常數的倒數 (73)，平均是x_{t-1}項常數除以x_{t-1}^2項常數的結果 (74)。

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{73}$$

$$\tilde{\mu}_t = \left(\frac{\sqrt{\alpha_t}}{\beta_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{74}$$

再將 (54)帶入至 (74)可以得到欲求分布之平均的表達式

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z\right) \tag{75}$$

附錄4-4. 推導擴散模型之損失函數

根據文獻，擴散模型之損失函數如 (8)、模型訓練的逆向擴散過程之分布如 (7)。模型要使預測分布的平均μ_θ逼近$\tilde{\mu}_t$，故μ_θ可表達成下式。

$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t)\right) \tag{76}$$

其中 x_t 與 t 都是訓練之輸入資料，分別代表雜訊圖與擴散時間。

因此可以計算μ_θ與$\tilde{\mu}_t$的誤差，我們使用 MSE，其中 C 為常數：

$$L_{t-1} - C = \mathbb{E}_{x_0,z}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}_t(x_t|x_0) - \mu_\theta(x_t, t)\right\|^2\right] \tag{77}$$

$$L_{t-1} - C = \mathbb{E}_{x_0,z}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z\right) - \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t)\right)\right\|^2\right] \tag{78}$$

使用分配率合併公式中的項目。先將$\frac{1}{\sqrt{\alpha_t}}$後面的部分合併再將$\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}$提出來，所以公式會變成：

$$L_{t-1} - C = \mathbb{E}_{x_0,z}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\sigma_t^2}\left\|(z - z_\theta(x_t, t))\right\|^2\right] \tag{79}$$

最後根據論文作者所述，將前面常數項移除可以得到比較好的結果；然後把 (52)帶入至 (79)。故最後的損失函數會變成如下所示，也是公式 (8)的完整推導過程。

$$L_{simple}(\theta) = \mathbb{E}_{t,x_0,z}\left[\left\|z - z_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z, t\right)\right\|^2\right] \tag{80}$$

附錄5.