Intro. to Artificial Intelligence Final Project

Team2

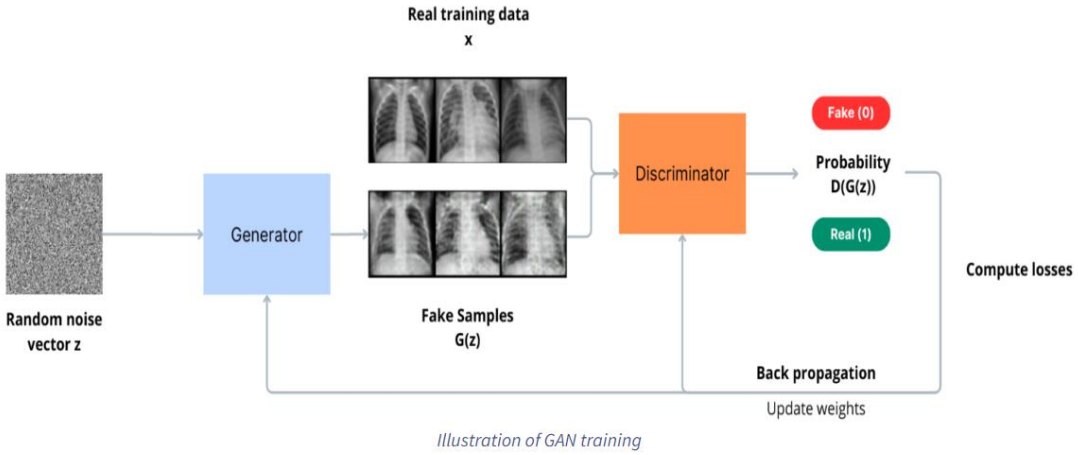
黃文彥 109550141、黃為碩 110950014、黃韋傑 109550056

1. Introduction

我們組進行的 Final Project 是以生成對抗網絡（GAN）為核心的高品質圖像生成專案。主要目標是開發一個能夠生成高品質圖像之模型，專門用於生成動漫人物臉部圖像，或真實數據之人臉圖像，希冀能透過生成對抗網路，也就是 GAN 來完成此效果，我們組認為生成臉部圖像的這種技術不僅擁有潛力改變自媒體或資訊產業，例如在遊戲角色設計、動漫、電影情境設計和虛擬現實(VR、AR)中創建更逼真的人物，還能在訓練數據受限的情況下創造出多樣化的數據集，用於進一步的機器學習和研究。

2. Literature Review / Related work

https://blog.ovhcloud.com/understanding-image-generation-beginner-guide-generative-adversarial-networks-gan/



截取自文章中 GAN 介紹

此篇文章研究介紹了生成對抗網絡（GAN）的基礎概念和應用，特別強調了深度卷積生成對抗網絡（DCGAN）的改進對提升圖像生成質量的重要性。DCGAN 透過使用跨步卷積（鑑別器）和分數跨步卷積（生成器）、批量標準化、以及在生

成器中使用 ReLU 激活函數和鑑別器中使用 LeakyReLU 激活函數，有效改進了圖像的合成質量和訓練的穩定性。這篇文章對我們的動漫頭像生成專案具有重要參考價值，尤其是在實現更高品質影像合成和提高模型穩定性方面。我們的成品會在此基礎上進一步探索特定於動漫風格的改進方法，進行調整、訓練模型以捕捉動漫特有之細節。

3. Datasets

我們從 Huggingface 平台下載了一個包含動漫人物臉部圖像的數據集，並對這些圖像進行了預處理，包括將圖像統一縮放到 64x64 像素統一大小，並進行標準化，這個操作是為了適配我們的 GAN 模型。

4. Baseline

在這個專案中，我們的 Baseline 模型是基於生成對抗網絡（GAN），這是一種由兩部分組成的深度學習框架，包含生成器(Generator)和鑑別器(Discriminator)。以下我們將會介紹本基線模型的具體實現和訓練過程之詳細描述：

生成器 (Generator)

- 生成器的目標是創造出足以欺騙鑑別器的假圖像數據，在這邊我們預期生成之圖片為動漫人物的頭像。這個生成器接受隨機噪聲向量作為輸入，通過全連接層和卷積層，轉換成與真實圖像相同尺寸的輸出。
- 在訓練過程中，生成器學習如何映射隨機噪音到一個數據分佈內，這個分佈的效果與真實數據分佈越接近越好。

鑑別器 (Discriminator)

- 鑑別器的任務是區分輸入的圖像是來自於真實數據集還是生成器產生的假圖像。我們以圖像為輸入的卷積神經網絡，輸出一個機率值，用以表示圖像為真實的可能性。
- 在訓練過程中，我們希望鑑別器能夠達到學會識別生成器之輸出，並對真實圖像給出高機率，對生成圖像給出低機率的效果。

5. Main Approach

在我們的專案中，Main Approach 是透過上述在 Baseline 部分所提到的生成對抗網絡（GAN），以及我們的生成器(Generator)和鑑別器(Discriminator)，來以更有效地方式生成高品質之動漫人物臉部圖像。下面我們將會介紹所選擇的模型和演算法具體描述，包括如何應用對於我們專案的需求。

模型架構:

1. 生成器 (Generator)
    - 輸入: 生成器的輸入是一個從高斯分佈中隨機抽取的噪聲向量，維度為 zzz。
    - 架構: 包括轉置卷積層（transposed convolution layers），每個層之間使用批量標準化（batch normalization）和 ReLU 激活函數。最後一層使用 tanh 作為激活函數，以生成與真實圖像同尺寸的輸出。
    - 輸出: 生成的圖像，尺寸與訓練數據集中的真實圖像相同。
    - 目標: 最大化鑑別器錯誤率，即讓鑑別器將假圖像錯判為真實。
    - 操作: 固定鑑別器參數，更新生成器的參數以產生能夠欺騙鑑別器的圖像。
    -
    -
2. 鑑別器 (Discriminator)
    - 輸入: 圖像（生成的動漫人物頭像或真實的動漫人物頭像）。
    - 架構: 包括數個卷積層，我們用於提取圖像特徵，每個卷積層後連接批量標準化和 LeakyReLU 激活函數。最後通過全連接層輸出一個機率值，用以表示輸入圖像為真實的可能性。
    - 輸出: 一個介於 0 到 1 之間的單一概率值，表示圖像是真實的概率。
    - 目標: 最大化對真實圖像正確標記為「真實」和生成圖像正確標記為「假」的能力。
    - 操作: 對於一批真實圖像和一批生成圖像，鑑別器更新其權重以提高區分真假圖像的準確性。

算法描述:

生成器 Generator

這個類繼承自 torch.nn.Module，用於建構我們的生成器模型。生成器的輸入是一個形狀為 (batch, in_dim) 的噪音向量，輸出則是一個形狀為 (batch, 3, 64, 64) 的圖像，其中 3 表示圖像的通道數（RGB），64x64 為圖像尺寸。

__init__(self, in_dim, feature_dim=64)

參數:in_dim: 噪聲向量的維度、feature_dim: 特徵維度的基數，用於控制各卷積層的深度。

層次結構

1. 第一層 self.l1
    - 功能: 將噪音向量映射到更高維度的特徵空間。
    - 部件:
        - nn.Linear(in_dim, feature_dim * 8 * 4 * 4, bias=False): 全連接層，將輸入維度 in_dim 映射到 feature_dim * 8 * 4 * 4。不使用偏置項來減少參數量。
        - nn.BatchNorm1d(feature_dim * 8 * 4 * 4): 一維批次標準化，用於穩定學習過程並加速收斂。
        - nn.ReLU(): 非線性激活函數，增加模型的非線性能力並幫助梯度更好地流動。
2. 第二層 self.l2
    - 功能: 逐步將特徵圖的維度上升，同時增加其空間尺寸（高和寬）。
    - 部件: 由三個 dconv_bn_relu 模組組成，每個模組都是一個轉置卷積層，用於將特徵圖的尺寸加倍。
        - 每個 dconv_bn_relu(in_dim, out_dim) 包括：
            - nn.ConvTranspose2d: 轉置卷積層，用於上採樣特徵圖，增加其尺寸。
            - nn.BatchNorm2d: 二維批次標準化。
            - nn.ReLU(True): 非線性激活函數。
3. 第三層 self.l3
    - 功能: 生成最終的圖像。
    - 部件:

        - nn.ConvTranspose2d(feature_dim, 3, kernel_size=5, stride=2, padding=2, output_padding=1, bias=False): 最後一個轉置卷積層，將特徵圖轉換成三通道的圖像。
        - nn.Tanh(): 激活函數，將輸出值壓縮到 [-1, 1] 範圍內，符合圖像像素的標準範圍。

forward(self, x)

- 輸入: 噪聲向量 x。
- 流程:
    - x 通過第一層 l1 轉換為特徵圖。
    - 特徵圖重塑為合適的維度以供後續層使用。
    - 經過三層轉置卷積處理後生成最終圖像。
- 輸出: 生成的圖像 y。

鑑別器 Discriminator

這個類同樣繼承自 torch.nn.Module，用於構建我們的鑑別器模型，該模型的輸入是一個圖像，輸出是一個表示圖像為真實的機率值。

__init__(self, in_dim, feature_dim=64)

參數:in_dim: 輸入圖像的通道數，RGB 圖像為 3、feature_dim: 特徵維度的基數，控制卷積層中的特徵數目。

層次結構

1. 第一層 self.l1
    - 功能: 逐層降低圖像的空間尺寸，同時提高特徵的維度，從而能夠從圖像中提取越來越抽象的特徵。
    - 部件:
        - nn.Conv2d(in_dim, feature_dim, kernel_size=4, stride=2, padding=1): 使用卷積層降低圖像的空間尺寸，同時從 3 個通道擴展到 feature_dim 個特徵。
        - nn.LeakyReLU(0.2): 使用帶有負斜率的 LeakyReLU 激活函數，我們認為這有助於維持模型的穩定性和梯度流。
        - 接續多層 conv_bn_lrelu 來進一步處理和深化特徵。

2. 自定義卷積-標準化激活 conv_bn_lrelu(in_dim, out_dim)
    - 功能: 進一步加深和細化特徵，並減少圖像的空間尺寸。
    - 部件:
        - nn.Conv2d(in_dim, out_dim, 4, 2, 1): 卷積層進一步降低空間尺寸，增加特徵維度。
        - nn.BatchNorm2d(out_dim): 批次標準化，用於正則化並加快訓練速度。在使用 WGAN-GP 時需替換為 nn.InstanceNorm2d。
        - nn.LeakyReLU(0.2): LeakyReLU 激活函數。
3. 卷積層和激活
    - nn.Conv2d(feature_dim * 8, 1, kernel_size=4, stride=1, padding=0): 最終卷積層，將特徵圖壓縮為單一特徵，無空間維度。
    - nn.Sigmoid(): Sigmoid 激活函數，將輸出壓縮至 0 和 1 之間，表示圖像為真實的概率。在使用 WGAN 時應移除此層。

forward(self, x)

- 輸入: 圖像 x，形狀為 (batch, 3, 64, 64)。
- 流程:
    - 圖像 x 經過第一層 l1 處理，提取特徵並逐步降低空間尺寸。
    - 最終輸出的特徵經過重塑，移除所有空間維度。
- 輸出: 機率值 y，形狀為 (batch)，表示每個輸入圖像為真實圖像之機率。

6. Evaluation Metric

我們使用多種方式作為 Evaluation Metric，包含主觀圖像之品質和真實性視覺檢查，以及量化方面的鑑別器準確率來評估模型的表現。我們將會先介紹用於評估生成器和鑑別器性能的主要量化指標，再介紹質化指標：

生成器和鑑別器的損失量化指標:

- 生成器損失:
  - 生成器的目標是最大化鑑別器犯錯的機率。當使用交叉熵損失時，生成器損失可以表示為：

$$L_G = -\frac{1}{N}\sum_{i=1}^{N}\log(D(G(z_i)))$$

  - 其中$G(z_i)$是生成器生成的圖像，$D(G(z_i))$是鑑別器對這些生成圖像的預測真實性概率。
- 鑑別器損失:
  - 鑑別器的目標是正確區分真實圖像和生成圖像。其損失可以表示為：

$$L_D = -\frac{1}{N}\sum_{i=1}^{N}[\log(D(x_i)) + \log(1 - D(G(z_i)))]$$

  - 其中 x_i 是真實圖像。

Inception Score (IS)

- Inception Score 用於評估生成圖像的質量和多樣性。它使用預訓練的 Inception 模型來計算生成圖像的條件標籤分佈和邊緣標籤分佈之間的 KL 散度：$IS = \exp(\mathbb{E}_x D_{KL}(p(y|x)||p(y)))$
- 其中 $p(y|x)$是給定生成圖像時標籤的條件分佈，$p(y)$是標籤的邊緣分佈。

質化指標部分，主要通過人為主觀進行觀察、我們將會視覺檢查生成圖像的真實性和細節，評估生成器的性能。包括查看生成之動漫人物圖像是否有任何明顯的缺陷如模糊、噪點或不自然的圖案等。此外我們也會進行穩定性測試，在長時間的訓練過程中，監測模型是否出現模式崩潰（mode collapse），代表生成器反覆生成高度相似或完全相同的圖像，藉此來檢測生成器和鑑別器之效能。

7. Results & Analysis

目前我們所訓練的動漫人物頭像生成模型已經能夠生成基本的圖像，也展示了生成器和鑑別器的性能狀況。我們的訓練週期為 1Epoch，學習率為 1e-4，Batch 大小 64，以質化評估來看，已經可以很明顯看出生成之圖片為動漫人物頭像，人臉輪廓、頭髮、眼睛的顯示較為完整，但在其他五官上則較無法辨識，顏色上多為重複且含有躁點，這可能是由於在上述提到的模式崩潰（mode collapse），也就是代表生成器反覆生成高度相似或完全相同的圖像所造成的結果。

8. Error Analysis

根據我們在 Results & Analysis 所提交的訓練結果截圖來看，生成的動漫頭像在細節上顯示模糊且含有躁點，特別是臉部特徵如嘴巴、鼻子與耳朵不夠清晰或不夠自然，此外圖片多為重複，過於一致，缺乏多樣性，生成器開始重複產生少數影像，最後就是部分影像出現顏色飽和度高或顏色不自然的情，下面將會根據這三個問題進行分析和改進策略:

對於五官模糊和含有躁點問題，可能原因因為生成器網路在處理細節方面的學習還不夠深入，或是因為模型訓練的不充分所導致的。改進策略包括增強資料集：使用更多高品質的動漫頭像進行訓練，確保資料集中包含豐富的樣式和特徵、調整模型架構：考慮引入或增加訓練模型中的殘差、注意力機制，我們認為這些可以幫助模型更好地捕捉和產生複雜的細節。

樣式一致性問題，可能原因為上述提及的模式崩潰現象，改進策略可以引入雜訊和變異，在訓練過程中為生成器輸入添加更多的雜訊變異，或使用不同的初始化策略，我們認為也可以使用多樣性損失：引入多樣性損失（如最小化生成影像間的相似度），激勵生成器探索新的影像空間。

最後就是顏色失真問題，產生這樣的問題可能是由於模型在色彩空間的表示學習不足，或是訓練資料中顏色分佈的偏差。改進策略包括調整顏色處理，在預處理階段對訓練影像進行顏色標準化，或在損失函數中加入顏色一致性的限制，也可以增加色彩多樣性：確保訓練資料中包含多樣化的顏色表達，以提升模型對不同顏色表現的學習能力。

9. Future Work

此專案繼續的話，我們會先完成上述在 Error Analysis 所提到問題之改進，解決掉躁點模糊、模式崩潰和色調統一的問題後，我們還想進行以下嘗試: 第一是增強模型架構，我們希望引入高階特徵提取，考慮採用更先進的捲積網路架構，像是 ResNet 或 DenseNet，我們認為這些架構應該能夠更有效地捕捉影像深層的細節和特徵。此外我們也會嘗試優化訓練過程，改進訓練演算法，嘗試不同的最佳化演算法像是 AdamW 或使用學習率衰減策略，有可能有助於模型更穩定地收斂。

關於資料處理部分，我們會考慮進行動態資料增強，即時地對訓練影像套用變換（如隨機裁切、旋轉、色彩調整），增強模型對不同視角和風格的適應性。最後就是提升生成品質的部分，我們認為可以透過多重任務學習，同時訓練模型進行頭像生成和其他相關任務（如風格分類），可以幫助模型更好地理解動漫風格的本質。

10. Code

https://github.com/shuo09/AI-Final-Team02/tree/main

Github 內包含完整程式碼、與 package requirements、模型訓練參數等等。

11. Contribution of each member

黃文彥 109550141 - 查找資料集、訓練模型，評估訓練結果，優化模型結果

黃為碩 110950014 - 查找可用模型、訓練模型、編寫報告、優化模型結果

黃韋傑 109550056 - 查找可用模型、訓練模型、評估訓練結果

12. References

https://blog.ovhcloud.com/understanding-image-generation-beginner-guide-generative-adversarial-networks-gan/

https://aws.amazon.com/tw/what-is/gan/

https://www.youtube.com/watch?v=4OWp0wDu6Xw

https://medium.com/ai-blog-tw/generative-adversarial-network-gan-%E7%9A%84%E5%9F%BA%E7%A4%8E%E7%90%86%E8%AB%96-503d8dc5ae79