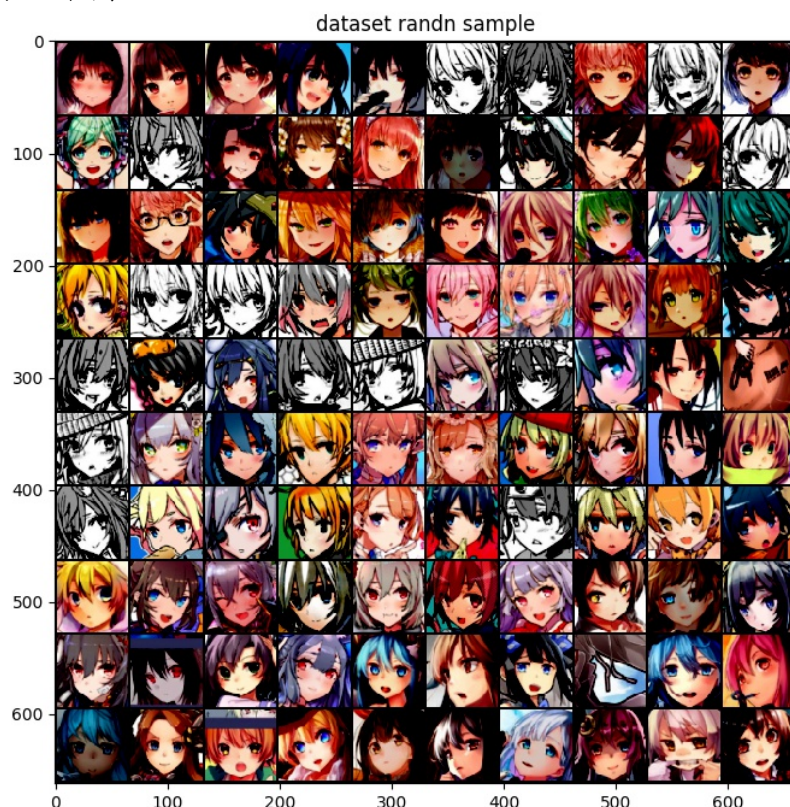


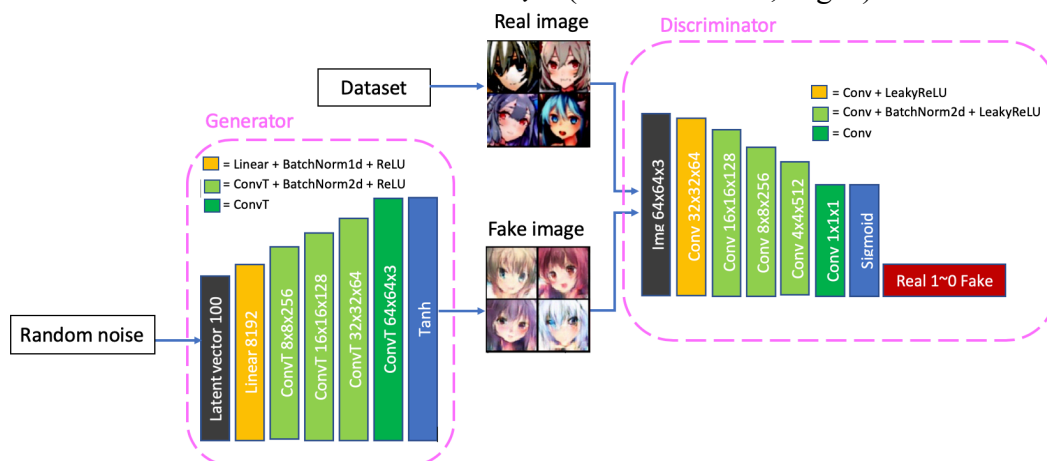
1. (2.5%) 訓練一個 model。

- a. (1%) 請描述你使用的 model (可以是 baseline model)。包含 generator 和 discriminator 的 model architecture、loss function、使用的 dataset、optimizer 參數、以及訓練 step 數 (或是 epoch 數)。

Dataset 為 71314 張 96x96x3 的二次元妹子圖片, batch size = 64, sampling 前 100 張如下圖：



Discriminator & Generator architecture 為：(baseline model, dcgan)



$$\text{Disc Loss} = BCELoss(D(img_{real}), label1) + BCELoss(D(img_{fake}), label0)$$

$$\text{Gen Loss} = BCELoss(D(img_{fake}), label1)$$

Optimizer 為 Adam, lr = 1e-4, betas=(0.5, 0.999)

b. (1.5%) 請畫出至少 16 張 model 生成的圖片。

下方為隨機生成的 16 張圖片，是在第 9 個 epoch 取的，randn seed 為 34。

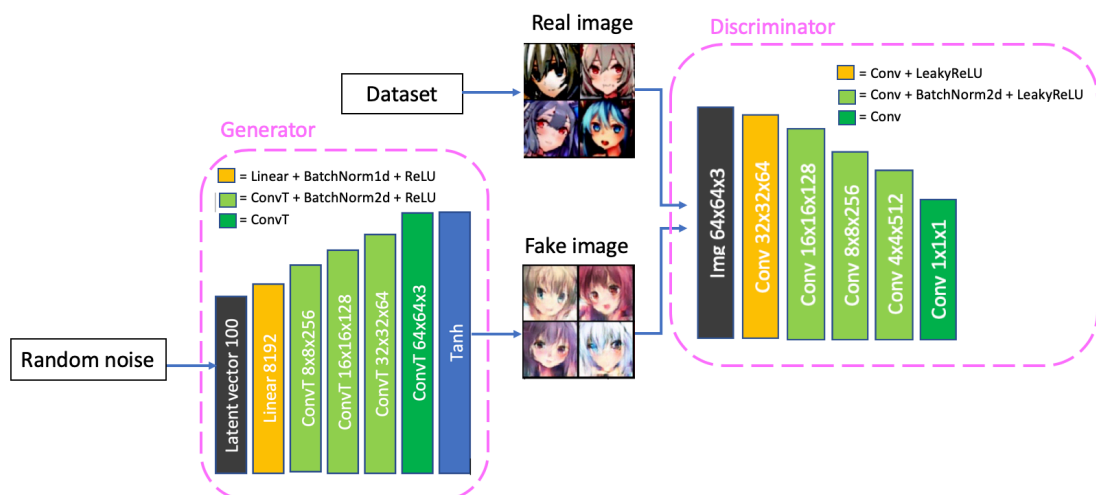


2. (3.5%) 請選擇下列其中一種 model： WGAN, WGAN-GP, LSGAN, SNGAN (不要和 1. 使用的 model 一樣，至少 architecture 或是 loss function 要不同)

a. (1%) 同 1.a，請描述你選擇的 model，包含 generator 和 discriminator 的 model architecture、loss function、使用的 dataset、optimizer 參數、及訓練 step 數 (或是 epoch 數)。

Dataset 為 71314 張 96x96x3 的二次元妹子圖片, batch size = 64, 同第一題。

Discriminator & Generator architecture 為：



$$\text{Disc Loss} = -\text{mean}(D(\text{img}_{\text{real}})) + \text{mean}(D(\text{img}_{\text{fake}})) + \lambda * \text{gradient_penalty}$$

$$\text{Gen Loss} = -\text{mean}(D(\text{img}_{\text{fake}}))$$

$$\text{Gradient Penalty} = \text{mean}((\|\nabla D(\text{img}_{\text{interpolate}})\|_2 - 1)^2)$$

$$\lambda = 250, n_{\text{critic}} = 3$$

Optimizer 為 Adam, lr = 20*1e-4, betas=(0.5, 0.999)

b. (1.5%) 和 1.b 一樣，就你選擇的 model，畫出至少 16 張 model 生成的圖片。

下方為隨機生成的 16 張圖片，是在第 39 個 epoch 取的，randn seed 為 4。



- c. (1%) 請簡單探討你在 1. 使用的 model 和 2. 使用的 model，他們分別有何性質，描述你觀察到的異同。

首先 wgangp 生成的影像清晰度稍微比 dcgan 來的偏模糊一點點點，因為前者是屬於 wgan 系列的 loss 的算法完全不一樣，以結果來看 dcgan 的 minimize binary cross entropy loss (=maximize JS divergence) 看起來會使圖片在清晰度上有比較好的表現，但 wgangp 的訓練穩定度就比 dcgan 來的好一點，使用的是 wasserstein distance (with $1 - Lipschitz$ weight clipping)，多勸幾個 epoch 也不會導致 model collapse，詳細結果可見第三題。而圖片的品質部分，兩個 model 生成的妹子在隨機 sample 下都有眼睛，都沒有到整個爛掉，可愛的也很可愛，我個人是覺得沒有差太多。

3. (4%) 請訓練一個會導致 mode collapse 的 model。

- a. (1%) 同 1.a，請描述你選擇的 model，包含 generator 和 discriminator 的 model architecture、loss function、使用的 dataset、optimizer 參數、及訓練 step 數（或是 epoch 數）。

使用與第一題的 model (dcgan) 與 parameter，詳見第一題，取第 18 epoch：17ep (如下圖，尚未 collapse)，18ep (如下圖，已完全 collapse)



- b. (1.5%) 請畫出至少 16 張 model 生成且具有 mode collapse 現象的圖片。下方為隨機生成的 16 張圖片，是在第 18 個 epoch 取的，randn seed 為 4。

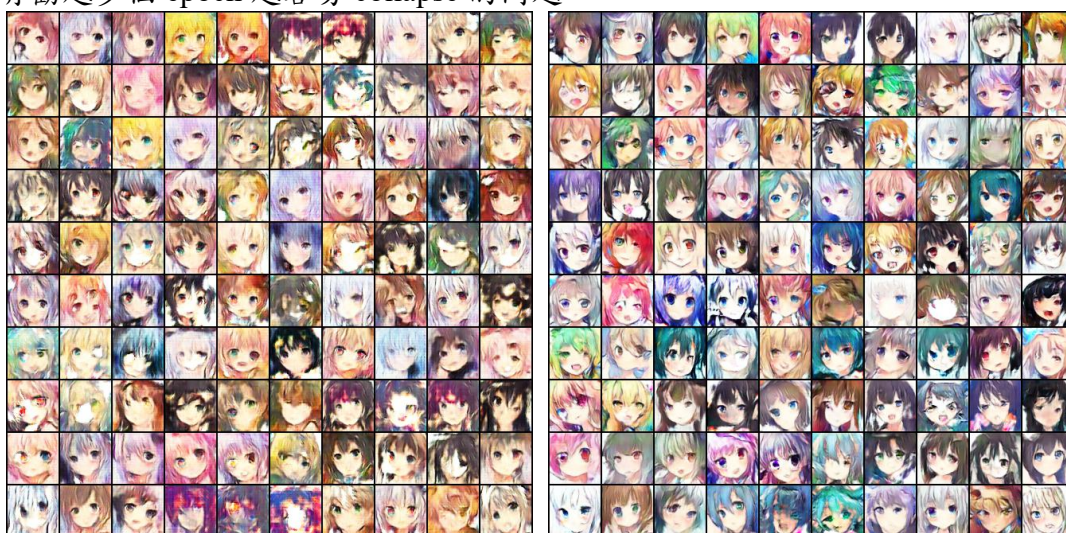


- c. (1.5%) 在不改變 optimizer 和訓練 step 數的情況下，請嘗試使用一些方法來減緩 mode collapse。說明你嘗試了哪些方法，請至少舉出一種成功改善的方法，若有其它失敗的方法也可以記錄下來。

法一：使用 wgan 系列的 model 可以避免 collapse 的問題，如用第二題的 wgangp (左) 取一樣的 optimizer parameter 在第 18ep 下的比較原本 dcgan (右)



但由於第二題的 wgan_gp 要能勸起來的參數不太一樣，因此雖然沒有 collapse 但在同樣的參數下很模糊，因此調整了一下 wgan_gp 的參數成和第二題一樣的參數，ep18 的結果如下圖(左)，ep39 的結果如下圖(右)，基本上並不會像原本有勸越多個 epoch 越容易 collapse 的問題。



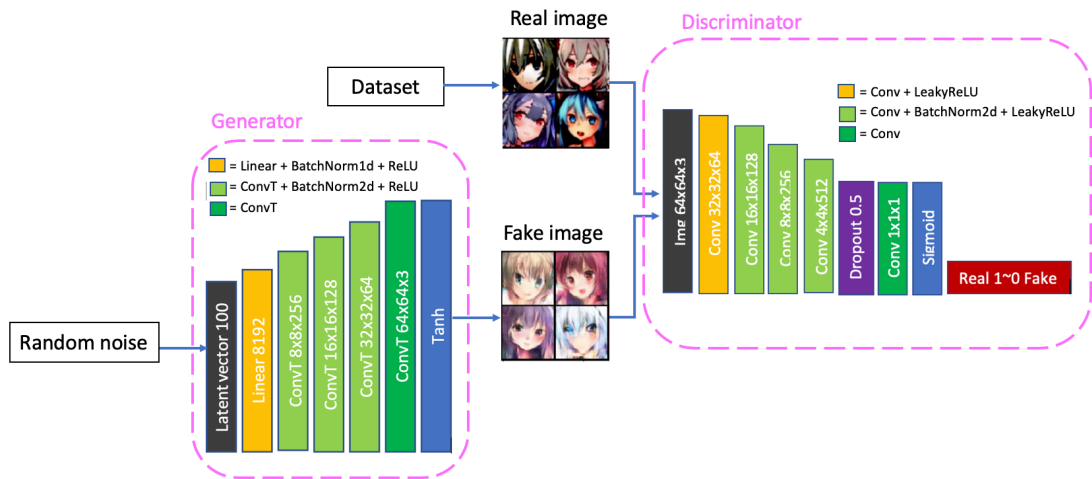
而這是因為 wgan 改進了 dcgan 等基本 gan 的假設，即假設 P_{data} 和 P_{gen} 這兩個高維空間中的低維 manifold 有重疊，因此可以用 maximize LS divergence (minimize binary cross entropy) 的方式求 loss，而 wgan 認為 P_{data} 和 P_{gen} 在高維空間中可能不會重疊，因此要用 wasserstein distance 來求 loss，效果就是能夠比較穩定的收斂而不會導致 collapse。

法二：用與原本第一題相同 model，相同的參數，但 Discriminator 的最後一個 Conv 層之前加一個 dropout=0.5 層，來減少 overfitting 的導致 collapse 的機會，一樣勸 18 個 epoch 結果如下(左)，勸 30 個 epoch 的結果如下(右)：

而其他嘗試如：把 dropout 也加到 generator 的最後一個 conv 之前，但這麼做會導致 model 勸不太起來，而只加入到 discriminator 會有幫助可以理解為，因為 discriminator 太 overfitting 於某種臉型，才會導致 generator 也致力於生成該種臉型導致 collapse，因此 dropout 僅需要加在 discriminator 即可解決問題。



Network architecture:



Raw figures:

