MLDS HW3 Report

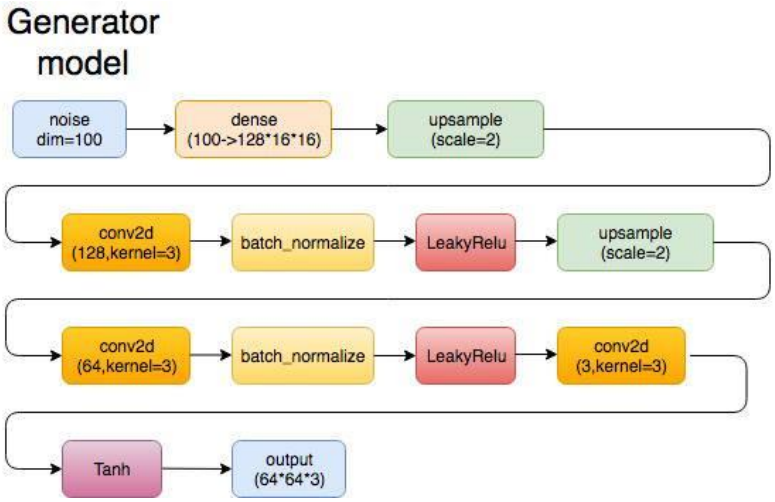
Image Generation

組員:

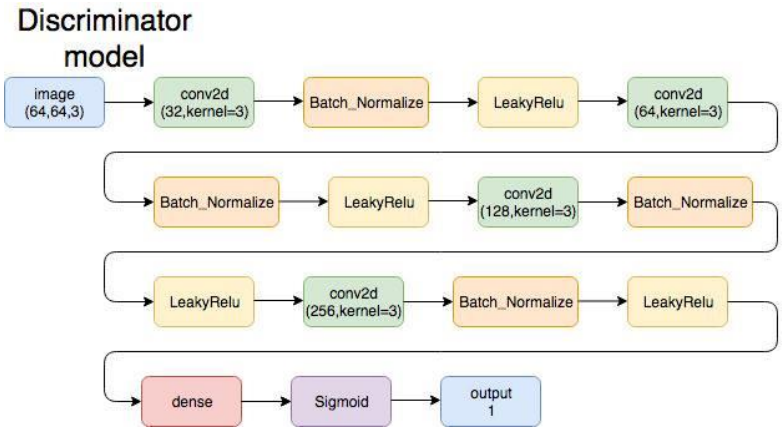
B03902093 張庭維 33% hw3-1 hw3-3
B03902101 楊力權 33% hw3-1 hw3-2
B03902102 廖廷浩 33% hw3-2 hw3-3

Model Description

- Image Generation:
  1. Generator Model:

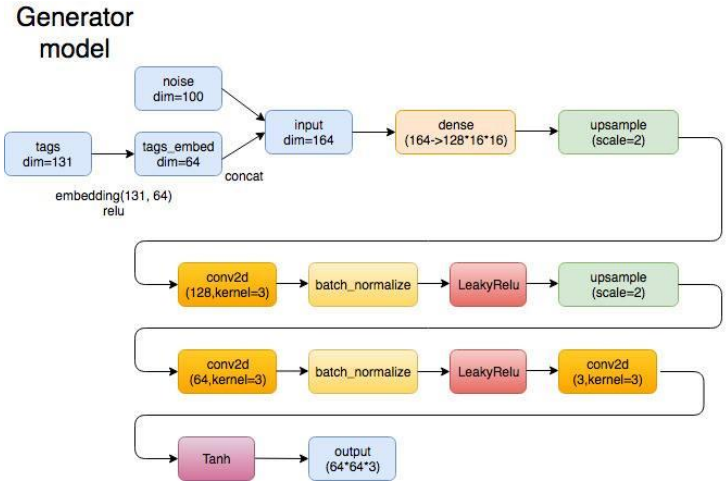


  2. Discriminator Model:

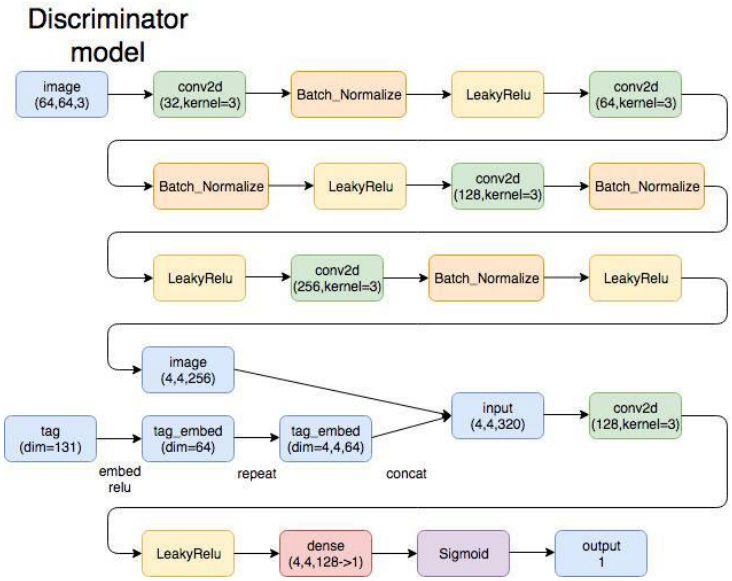


- Text-to-image Generation:

1. Generator Model:



2. Discriminator Model:



Experiment settings and observation

- Image Generation:

  Batch size = 128

DC-GAN (Best Result Observed):

With implementing tips 1, 3, 4, 6, 9 and the model structure mentioned above.
Epoch=500

Setting:

Optimizer: both adam with lr = 0.0002

Loss Function: BCError (with ground truth image labeled by 1 and the generated image labeled by 0)

Update rate: Generator : Discriminator = 1 : 5

Observation:

這是目前觀察下來表現的最好的 model，但還是會有五官崩壞的情況，下面會有我們做的其他種嘗試解決的方案，包括 WGAN、WGAN-GP、及灰階的 GAN 加上上色 Autoencoder Model 的 Hybrid Model

- Text-to-Image Generation

DC-GAN

Setting:

Optimizer: Adam with lr = 0.0002

Loss Function: BCError

Epoch:200

Update rate: Generator : Discriminator = 1 : 5

Observation:

對 tag 做 embedding 時，embed 大小跟 noise 的比例會影響影像的生成，當 noise(dim=100)，embed_size=128 時，不同的 noise 會生出長得非常相似的圖，然而當 noise(dim=100)，embed_size=64 時，在不同的 noise 下可以生成不一樣的臉型，表示在 tag embed_size 大的時候，往往會在圖片生成

上造成比 noise 更大的影響力，所以說使用適當的 embed_size 是一個生成圖片的關鍵，我們用 embed size 參數為實驗，結果如下圖；

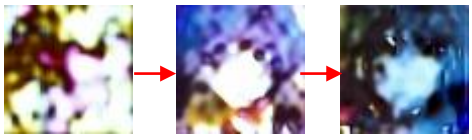
不同 noise，相同 tag，Embed_size=128


不同 noise，相同 tag，Embed_size=64


不同 noise，相同 tag，Embed_size=32


Compare your model with WGAN, WGAN-GP, LSGAN

1. **W-GAN**



Epoch 1　　Epoch 100　　Epoch 500

With implementing tips 1, 3, 4, 6, 9 and the model structure mentioned above, except the ‘sigmoid’ layer in the end.

Model Setting:

Model: use the same model in model description without the ‘sigmoid’ layer in the end

Optimizer: both adam with lr = 0.0002

Loss Function: Mean score of Discriminator(gan output)-Mean score of Discriminator(ground truth)

Update rate: Generator : Discriminator = 1 : 5

Other Setting: Using Weight Clipping for discriminator’s parameters with clipping value [-0.01, 0.01]

Analysis:

並沒有比 DC-GAN 優秀，而且訓練過程也遠較 dc-gan 緩慢，要到將近 100

epoch 才看的出一張相當模糊的人臉 (dc-gan 約 20~30 就可以有相近的結果)

2. **W-GAN GP**

Epoch 1 Epoch 30 Epoch 500

With implementing tips 1, 3, 4, 6, 9 and the model structure mentioned above, except the 'sigmoid' layer in the end.

Model Setting:

Model: use the same model in model description without the 'sigmoid' layer in the end

Optimizer: both adam with lr = 0.0002

Loss Function: Mean score of Discriminator(gan output)-Mean score of Discriminator(ground truth) + λ *Gradient Penalty

Update rate: Generator : Discriminator = 1 : 5

Other Setting: We Tried λ = 0.1 to 10

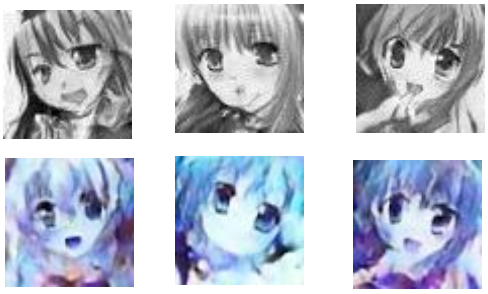
Analysis:

在訓練速度上，有接近 DC-GAN 的效果，可以在 20~30 epoch 看出模糊的人臉，但再繼續訓練下去，並沒有明顯的變好，而且色調不知為何都相當的偏暗。在 W-GAN 上我們做了比較多的實驗，主要是發現若不調整 Generator 與 Discriminator backward 次數的比例，使他小於 1:5，發現大多會直接 train 壞，可能是一開始 discriminator 學的太慢造成的。也 tune 過了 λ 的值，結果是 0.1 表現比較好。

Training tips for improvement

Origin Image at 500 epoch with the simplest model

1. **Hybrid Model (Gray-Scale GAN + Colorize Autoencoder Model)** tip 7



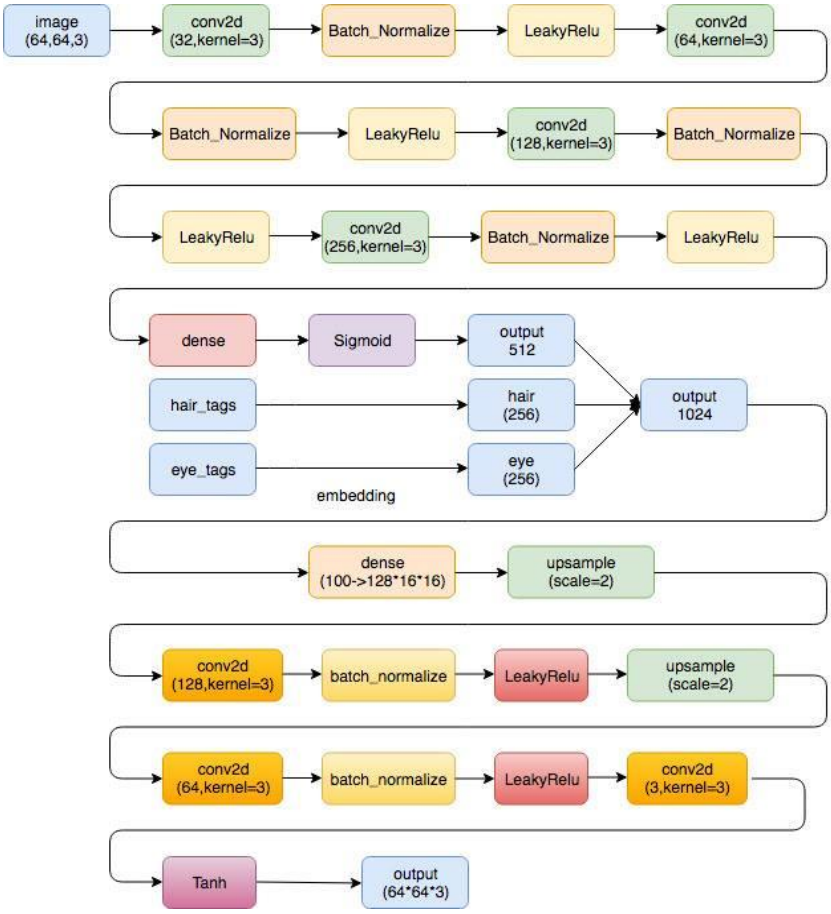
With label blue eyes and blue hair

Setting:

GAN Model: Same structure as the origin DC-GAN model, with then channel

modified to 1

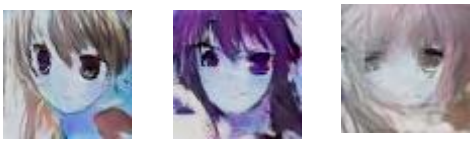
Colorize Model: An Autoencoder whose inputs are gray scale images generate by GAN model above and eyes color and hair color as condition. Structure shown below.



Analyze:

普遍來說 Gray Scale GAN 出來的圖片解析度會比直接 GAN 彩色的高些許，表現也普遍較好，但是 Colorize 之後又會糊掉，而且在 train 的時候是針對 Conditional 的情況下去做的，會出現一種問題是當深色的 Gray Scale 圖片硬要畫上一個淺色的髮色或眼睛顏色時，會直接壞掉。聽說對 Gray Scale 圖片做 normalize 再過這個 Network 會比較好，但由於時間緊迫就沒有做這個實驗了。

2. **Sample Noise by Gaussian Distribution tip 3**



With 500 epoch

Setting:

Generate Random Noise by Gaussian distribution.

Analyze:

有做這個 tip 感覺五官會柔合一點，但是效果並不十分顯著。

3. **Don't balance loss via statistics tip 11**

Setting and Analyze:

我們試著讓 G loss 大於 D loss 時，train D，而反過來時，train G，且發現效果奇差，基本上在 100 epoch 後會變成一片白色，因此我們比較專注在 tune 這個 statistic 的部分，發現小於 1:5 會使 model 壞掉，而 1:5 到 1:10 看起來 100 epoch 內結果相似，但 1:10 的略差。我們上面的 Best Result 就是用 1:5 train 出來的。

HW 3-3

Reference:

A Neural Algorithm of Artistic Style - By Leon A. Gatys, Alexander S. Ecker,

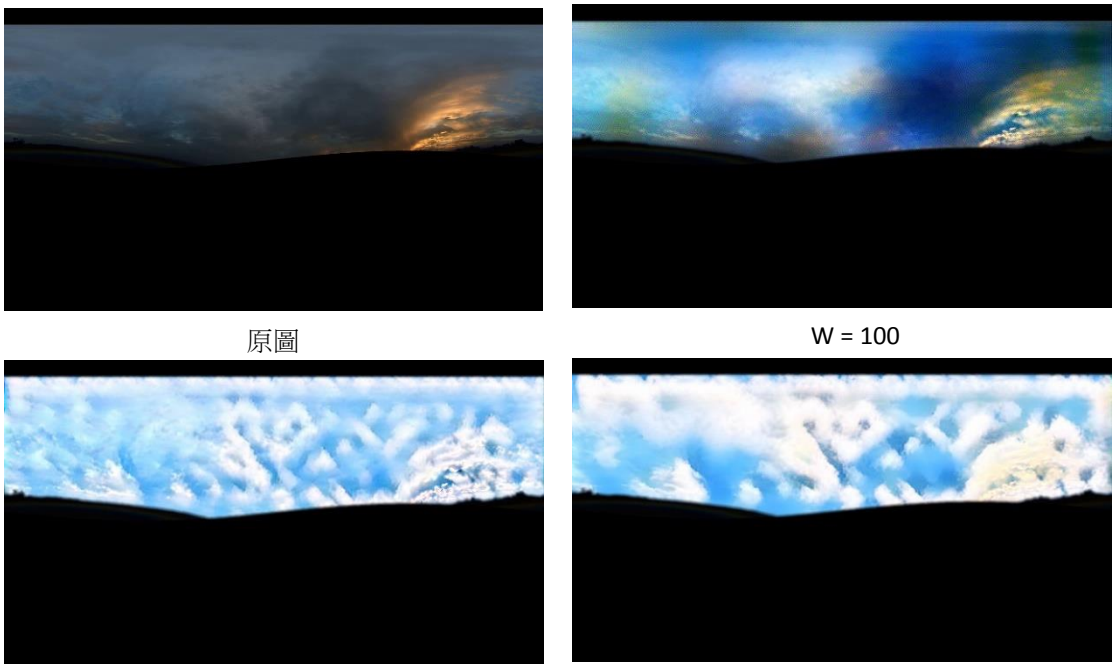
1. **Show our result**

將左上方的原圖，分別 style transfer 乘下面三個 style

2. **Analyze**
    1. 在 style weight/ content weight 的比例上我們有稍微調過，如下圖一，當比例(W)為 100、1000、10000 時結果分別如下圖
    2. Content layers 跟 Style layers 基本上預設的是 content 拿第 4 層，style 拿 1~5 層(用 pre-train 好的 VGG19 model)，這也是網路上查到做 style

transfer 最常用的參數，我們自己 tune 過之後，要不是變很爛，不然就是結果不明顯，因此維持預設值。

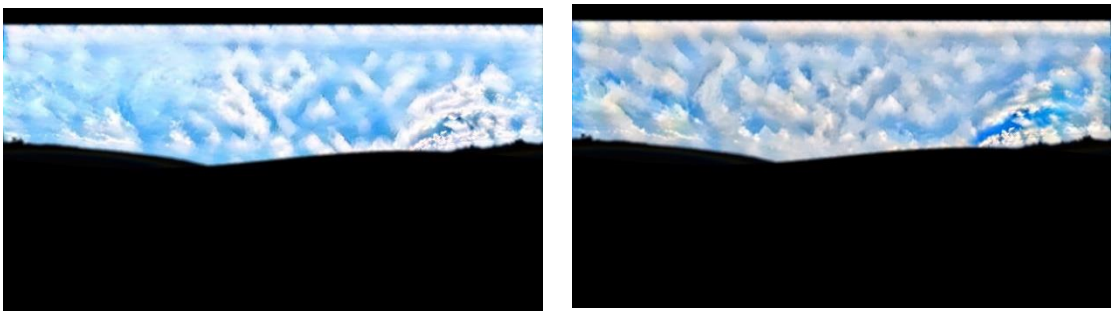
3. Img input 是一開始送進去 Style Transfer 的圖，預設是使用 content 的原圖，從它開始 train，我試過直接送白色的圖進去 train，從白色的圖開始畫，會發現在 sky2, sky3 時雲朵變得很塊狀，並不向真正的天空，可能是因為雲作為 content 的判定在陰天時，可能是一小塊一小塊的，若從白的開始 train，雲連在一起的現象電腦就看不出來了，因此雲會被切得很瑣碎，不太像真正的天空。圖(二)



原圖　　W = 100

W = 1000　　W = 10000

圖(一)



Start with origin image　　Start with white image

圖(二)