HW3

- HW3-1、HW3-2
- Model Description

a.Image Generation (2%)

Generator	Discriminator
Input=(100,) Reshape((1,1,100)) Conv2DTranspose k4n512s1 BatchNormalization LeakyReLU(0.2) Conv2DTranspose k4n256s2 BatchNormalization LeakyReLU(0.2) Conv2DTranspose k4n128s2 BatchNormalization LeakyReLU(0.2) Conv2DTranspose k4n64s2 BatchNormalization LeakyReLU(0.2) Conv2DTranspose k3n64s1 BatchNormalization LeakyReLU(0.2) Conv2DTranspose k4n3s2 tanh	Conv2D k4n64s2 LeakyReLU(0.2) Conv2D k4n128s2 BatchNormalization LeakyReLU(0.2) Conv2D k4n256s2 BatchNormalization LeakyReLU(0.2) Flatten Dense(1) sigmoid

generator_optimizer = Adam(lr=0.00015, beta_1=0.5)

discriminator_optimizer = Adam(lr=0.0002, beta_1=0.5)

epochs=10000

batch_size=64

b.Text-to-image Generation (2%)

Generator:

輸入為128維的noise(~uniform(-1, 1))，12維的頭髮(one-hot)跟11維的眼睛(one-hot)。

網路架構：

1. concat, dense(12*12*64), bn, relu, reshape([12, 12, 64])
2. 16個residual block
3. conv(64, 3), bn, relu, +1.的輸出
4. 3個upsampler
5. conv(3, 9), tanh

conv(a, b): a為filter數量，b為filter size(b*b) residual block: input=image, output=bn(conv(relu(bn(conv(image, 64, 3))), 64, 3)) + image

upsampler:input=image, output=relu(bn(pixel_shuffler(conv(image, 256, 3))))

Discriminator:

輸入為96*96*3的image

網路架構：

1. conv(32, 4, 2), leaky_relu, 2個residual block(32, 3)
2. conv(64, 4, 2), leaky_relu, 2個residual block(64, 3)
3. conv(128, 4, 2), leaky_relu, 2個residual block(128, 3)
4. conv(256, 3, 2), leaky_relu, 2個residual block(256, 3)
5. conv(512, 3, 2), leaky_relu, 2個residual block(512, 3)
6. reshape(3*3*512), fc1(2)(真假), fc2(12)(頭髮), fc3(11)(眼睛)

conv(a, b, c): a為filter數量，b為filter size(b*b), c為stride

residual block:

input=image, output=leaky_relu(seperable_conv(leaky_relu(seperable_conv(image))) + image)

Discriminator所有的變數都有spectral normalization

Generator Loss:

Minimize:

lambda*[fc1 cross entropy(**fake, real**)] + [fc2 cross entropy(hair)] + [fc2 cross entropy(eye)]

Discriminator Loss:

Minimize:

lambda*[fc1 cross entropy(**real, fake**)] + [fc2 cross entropy(hair)] + [fc2 cross entropy(eye)]

- Experiment settings and observation

a.Image Generation (1%)

經過Conv2DTranspose的generator比只用Conv2D的DCGAN穩定(如果太多epoch DCGAN會train不起來)、色彩鮮豔(雖然髮型顏色略有不同，但感覺整體趨向同一色系，用Conv2DTranspose的DCGAN是完整的髮型顏色變化)，且解析度較只有用fully connected layer的GAN高，用fully connected layer的GAN看起來有很多一點一點的雜訊，因為每個pixel是獨立處理，跟Convolution的network相比pixel之間的關聯較差。

產生結果：

b.Text-to-image Generation (1%)

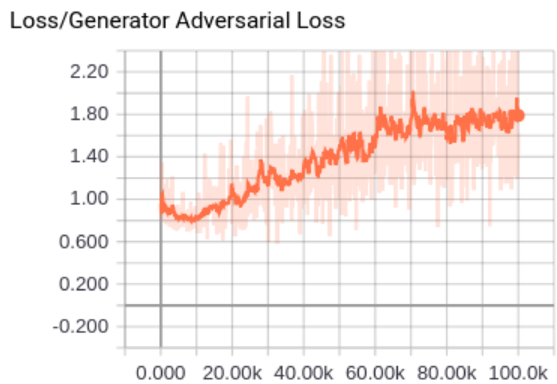
Experiment setting:

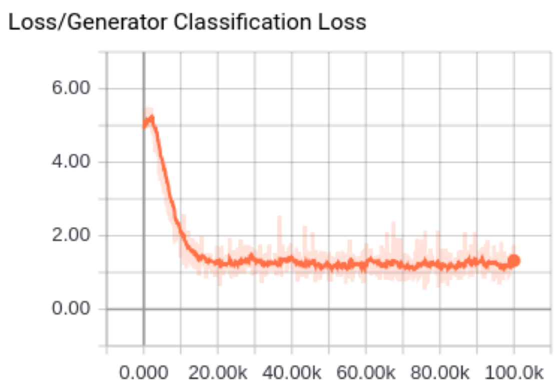
影像的輸入輸出皆為96*96*3

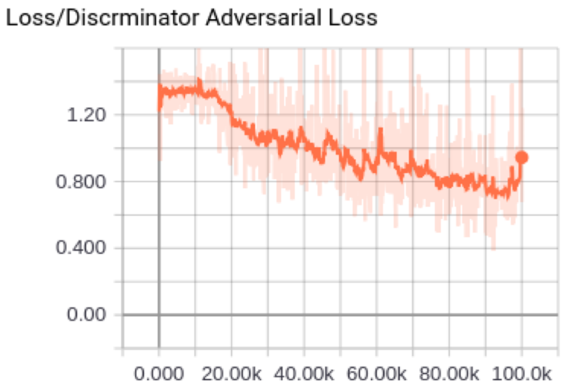
batch size=64, learning rate=2e-4, Adam Optimizer(beta1=0.5, beta2=0.999), lambda=25

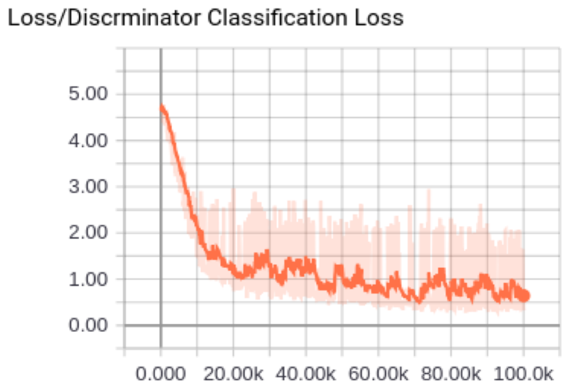
generator跟discriminiator各交互訓練一次，總共1e5個iteration

Training curve:

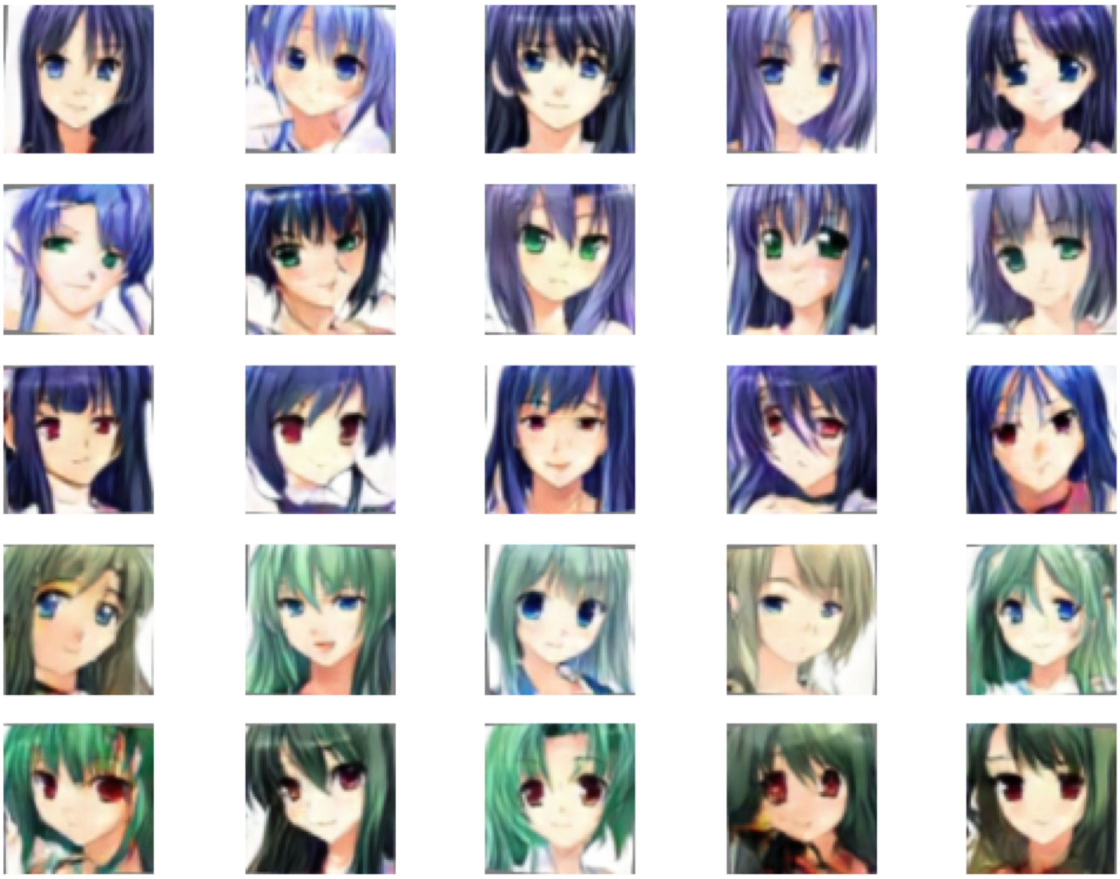





Loss/Discrminator Adversarial Loss
1.20
0.800
0.400
0.00
0.000
20.00k
40.00k
60.00k
80.00k
100.0k


Loss/Discrminator Classification Loss
5.00
4.00
3.00
2.00
1.00
0.00
0.000
20.00k
40.00k
60.00k
80.00k
100.0k

產生結果：



- Compare your model with WGAN, WGAN-GP, LSGAN (choose 1) (Image Generation Only)
- Model Description of the choosed model (1%)

  用WGAN，將Generator及Discriminator(critic)用上述的model架構(DCGAN with Conv2DTranspose)，根據WGAN的paper的algorithm，設定參數：learning rate=0.00005, weight clipping range=(-0.01,0.01), n_critic=5, optimizer=RMSprop。

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

- Result of the model (1%)

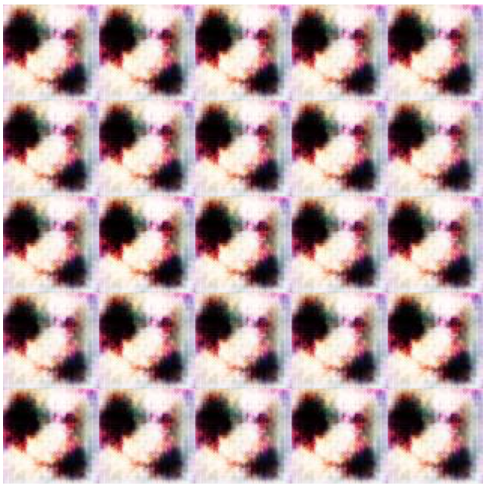
- Comparison Analysis (1%)

  WGAN在訓練上似乎需要更多epoch才能達到像DCGAN的成果，因為weight被bound住，所以跟原來的DCGAN相比，同樣經過10000個epoch但成果看起來沒有DCGAN好，有些臉仍是模糊的，且顏色也偏暗。

• Training tips for improvement (Image generation Only) (6%)

a.3 - Use a spherical Z

將np.random.normal改成np.random.uniform，其他情況不變，會完全train不起來，如下圖：

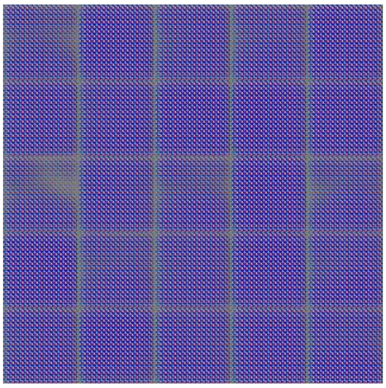


b.4 - BatchNorm

在只有Conv2D的DCGAN中加入BatchNorm，在train 1000個epochs後可以發現有BatchNormalize的GAN(下面右圖)較沒有BatchNormalize的GAN(下面左圖)顏色來的鮮豔，而且細節也較為清楚，所以加入BatchNormalize後可以train的較快，也不會太過偏向某個色系。

在一起train了5000個epochs之後的結果如下，下圖左邊為沒有BatchNormalize的結果，右圖為有BatchNormalize的結果，發現兩者差異不大，所以可以推測BatchNormalize只有在training前期影響較大。

將原model的BatchNormalize移除，其他情況不變，則完全train不起來，如下圖：

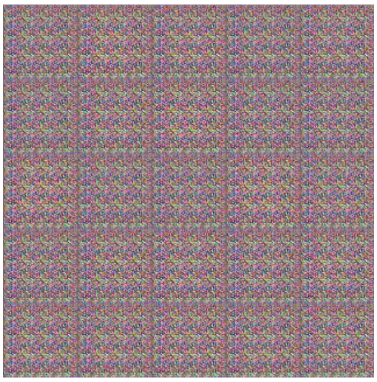


c.5 - Avoid Sparse Gradients: ReLU, MaxPool

將所有LeakyReLU改成ReLU，在只有Conv2D的DCGAN中會有色彩太過偏亮的問題，且圖片間的色彩同質性過高，如下圖左邊為ReLU，右邊為LeakyReLU。

在用Conv2DTranspose的DCGAN中把LeakyReLU改成ReLU完全train不起來。

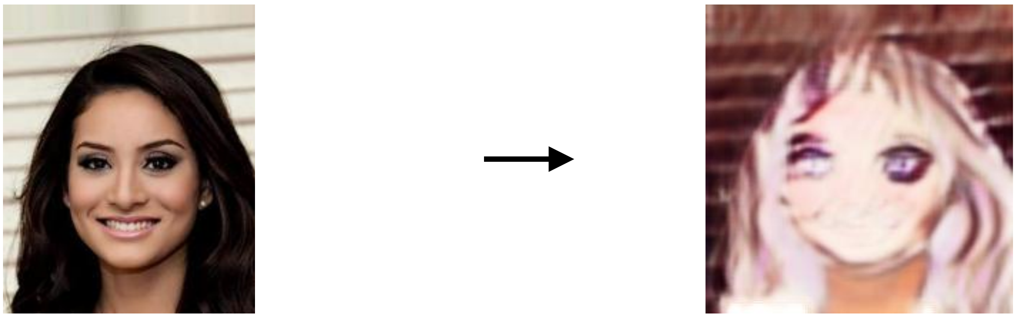
如下圖：



- HW3-3
- Show your result (1%)

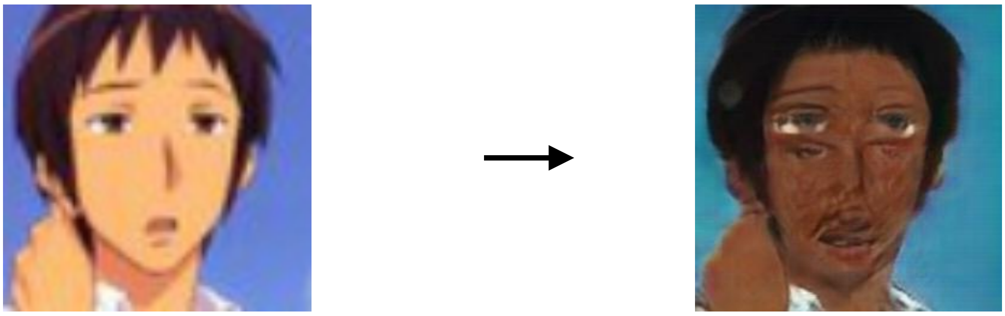
Two domain and transfer result.

1.真人照片轉換為動漫照



Sample結果：

2.動漫照轉換為真人照



Sample結果：

- Analysis (1%)

Which model you use and your observation.

1.使用Cycle GAN來進行風格轉換，而在DAataset的部分，真人domain使用CelebA、動漫domain使用作業三的動漫dataset。

2.訓練的過程中，發現的幾個有趣的地方：

a.比較dataset兩種風格各放入一千張與各放入10萬張來進行比較：

dataset為一千張照片：可以看出，基本上只是色調上的轉換，真人圖片帶有馬卡龍色，而動漫照片則開始出現負片感、顏色加深mm，因為真實照片基本上顏色較為深沉。

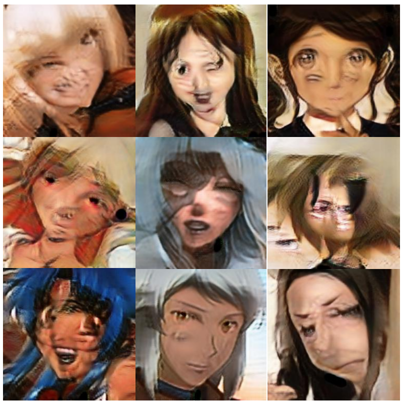
真人照片轉換為動漫照

動漫照轉換為真人照

dataset為十萬張照片：此時可以看出，較能夠做到真正的風格轉換，不僅止於色調上的變化，真人圖像轉為動漫時，大眼睛的特徵也會加入， b而在動漫轉真人的部分，光影的變化、鼻子輪廓加深、眼睛變小也都一一展現出來。

真人照片轉換為動漫照

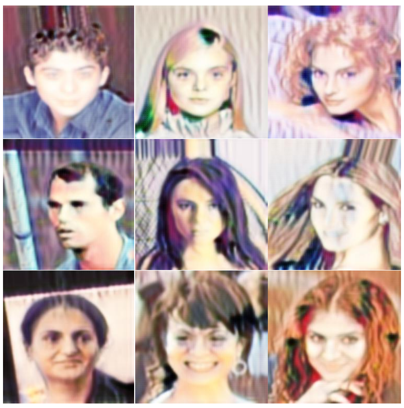
動漫照轉換為真人照



b.比較不同iteration後照片轉換差異（以dataset十萬張照片為基準）

1000個iteration：此時為最一開始，基本上為濾鏡的轉換，真人照片色調轉為馬卡龍色，而動漫照則顏色轉深，並多了光影。

真人照片轉換為動漫照



動漫照轉換為真人照

10000個iteration：此時，真人轉動漫照漸漸學習出若有大眼睛比較符合動漫domain，而動漫照則把大眼睛部分遮掩掉，或以較小的眼睛取代，鼻子部分也更加深輪廓。

真人照片轉換為動漫照

動漫照轉換為真人照

- 分工表

HW3-1 鄭雅文

HW3-2 楊碩碉

HW3-3 陳品君