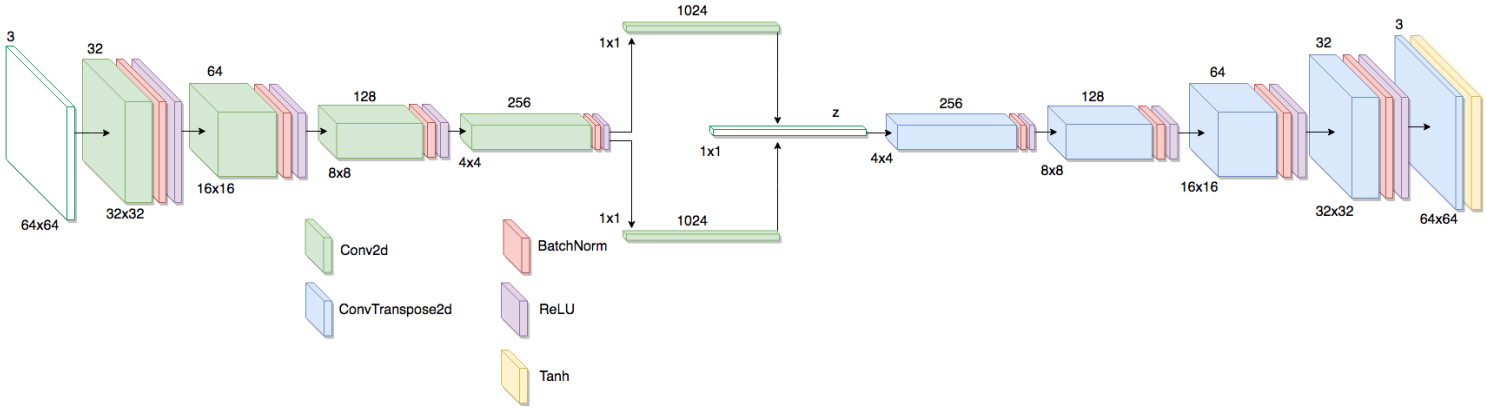
Deep Learning for Computer Vision　Hw4 Report

電機三　B04901096　蔡昕宇

Image Generation

1. Variational AutoEncoder (VAE)

1-1 Describe the architecture and implementation details of your model.



上圖為這次實作 VAE 的 model，由於架構類似 autoencoder 的架構，因此在實作中先架好 autoencoder 才把架構慢慢換成 VAE 的架構。以下更詳細的說明每一層的架構。

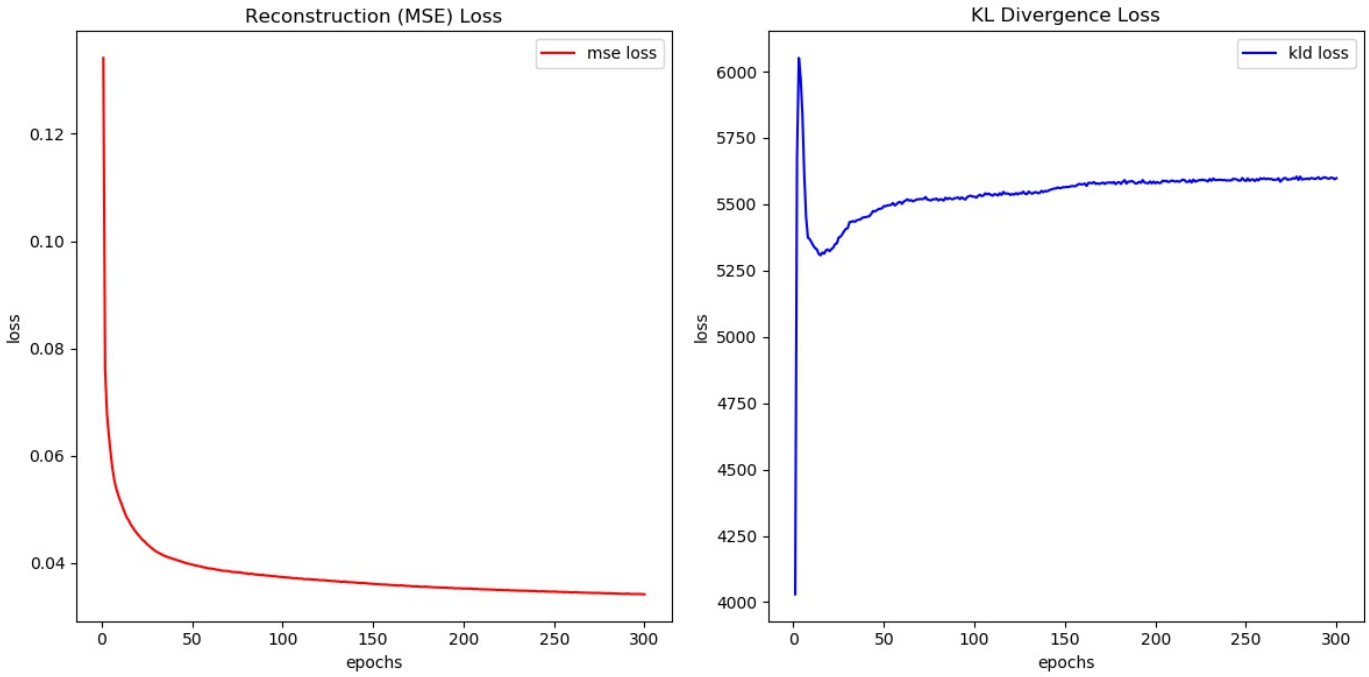
decoder / latent (z)	**encoder**
4 x 4 conv. 32 ReLU. stride 2	4 x 4 conv. 256 ReLU. stride 2
4 x 4 conv. 64 ReLU. stride 2	4 x 4 conv. 128 ReLU. stride 2
4 x 4 conv. 128 ReLU. stride 2	4 x 4 conv. 64 ReLU. stride 2
4 x 4 conv. 256 ReLU. stride 2	4 x 4 conv. 32 ReLU. stride 2
4 x 4 conv. 1024 ReLU. stride 1 (μ) 4 x 4 conv. 1024 ReLU. stride 1 (σ) output layer $z = \mu + \epsilon\sigma = \mu + e^{0.5 \cdot logvar}$	4 x 4 conv. 3 Tanh. stride 1

Input size 為 3 x 64 x 64，且 normalize 到 -1~1 之間，因此 output activation function 使用 Tanh。其中本架構沒有使用任何 fully connected 的 layer，全部為 convolutional layer，比較符合對於影像的處理。而 ϵ 是從 normal distribution 的 random variable，$\epsilon \sim N(0,1)$。這樣的目的使得原本的 random sampling 不可微分變得可以 backprop，這個過程稱之為 reparameterization trick。

在 loss function 使用 pixel-wise averaged mean square error (MSE) 和 KL divergence。而 MSE 則是為了 minimize reconstruction error，而 KL divergence 是為了讓 latent space 越像 $N(0,1)$ 越好。而為何要 map 到 normal distribution，則是因為給定 training data 的隨機分佈，都可以透過 linear operation 從 $N(0,1)$ map 到此一分佈。而這樣的 linear operation 可能是透過 dencoder 的前幾層實作這樣的 operation，因此認為可以 reconstruct 影像。[1]

[1] Tutorial of Variational Autoencoders: https://arxiv.org/pdf/1606.05908.pdf

1-2 Plot the learning curve (reconstruction loss and KL divergence) of your model



Loss function: $L_{reconstruction} + \lambda_{KL} L_{KL}$，$L_{reconstruction}$為 pixel-wise mean square error，而在 KL divergence 的部分，實作選定的 $\lambda_{KL} =$ 3e-6

Reconstruction loss 在 100 epochs 之後大約下降 0.03 左右，KL divergence 約在 50 epochs 之後大約維持在 5500 左右，reconstruct 的結果在 100 epochs 之後結果差不多。

1-3 Plot 10 testing images and their reconstructed result of your model and report your testing MSE of the entire test set.

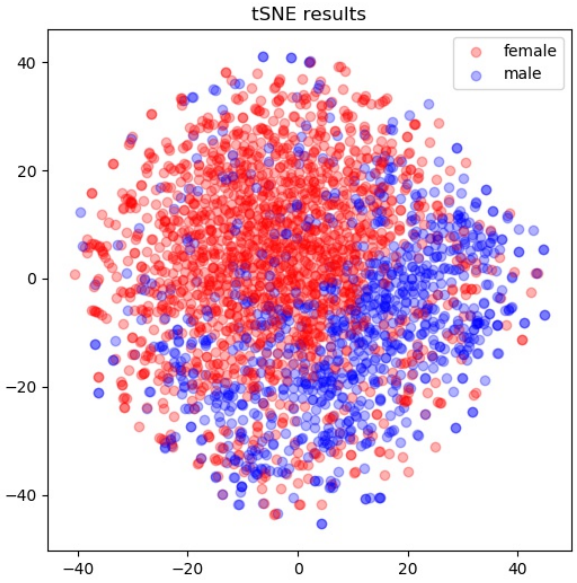
Reconstructed images (上排為 testing images，下排為 output 出來的 images)

Results: MSE of testing set: 0.08907

1-4 Plot 32 random generated images of your model.

1-5 Visualize the latent space by mapping test images to 2D space (with tSNE) and color them with respect to an attribute of your choice.

選用的 attribute 與投影片相同，為 Male/ Female，在圖中看到有被分到兩邊的樣子。



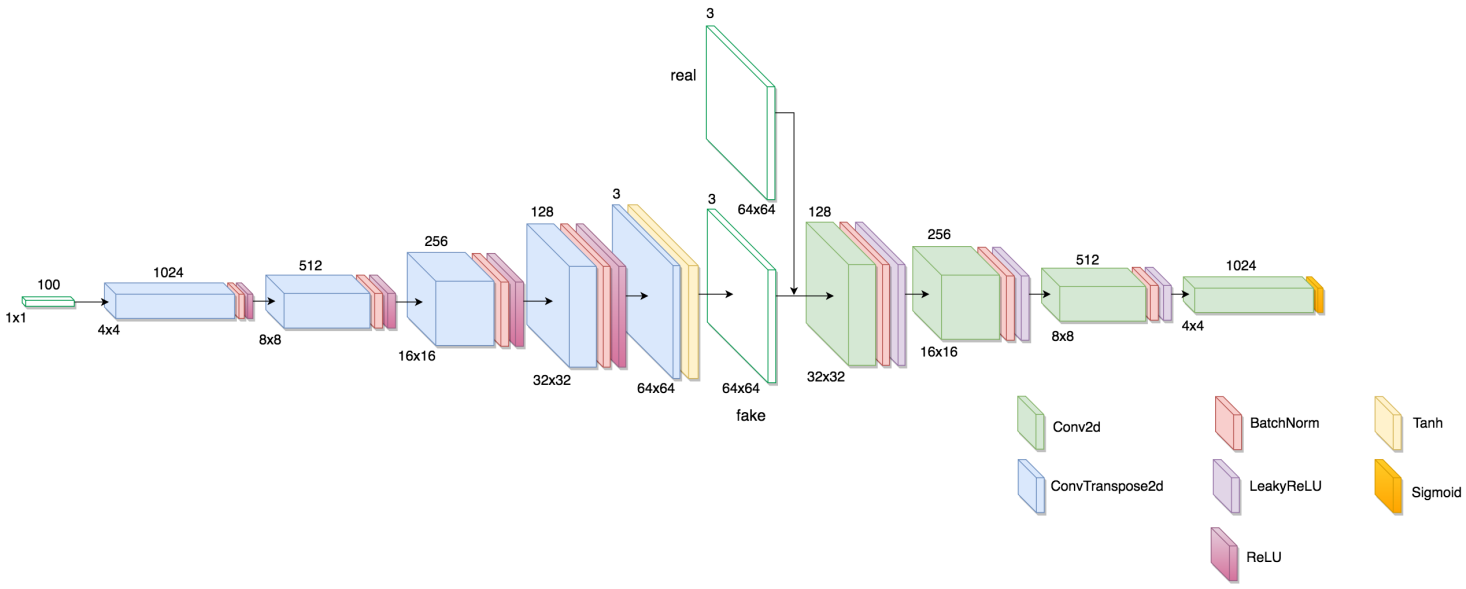
1-6 Discuss what you've observed and learned from implementing VAE.

在此實作中發現，調整 λ_{KL}對生成的結果的影響非常大。如果將 λ_{KL}調小，KL divergence 相對可以變大，因此可能早成 random generation 會因此出現更多雜訊。反之，如果調大，可能使得 reconstruction 變得較差一些。一開始 λ_{KL} = 1e-5，在實作中發現 reconstruction 沒有很理想，比較容易 reconstruct 出較模糊的影像，因此將 λ_{KL}調小。

而 VAE 的 reconstruction 較為模糊，random generation 也仍然是一個模糊的影像，可能是在 encoder 的架構中，latent space 可能取得只有局部重要的資訊，而並沒有影像大局的感覺，因此局部跟局部間的關係可能比較沒那麼緊密。

2. Generative Adversarial Network (GAN)

2-1 Describe the architecture and implementation details of your model

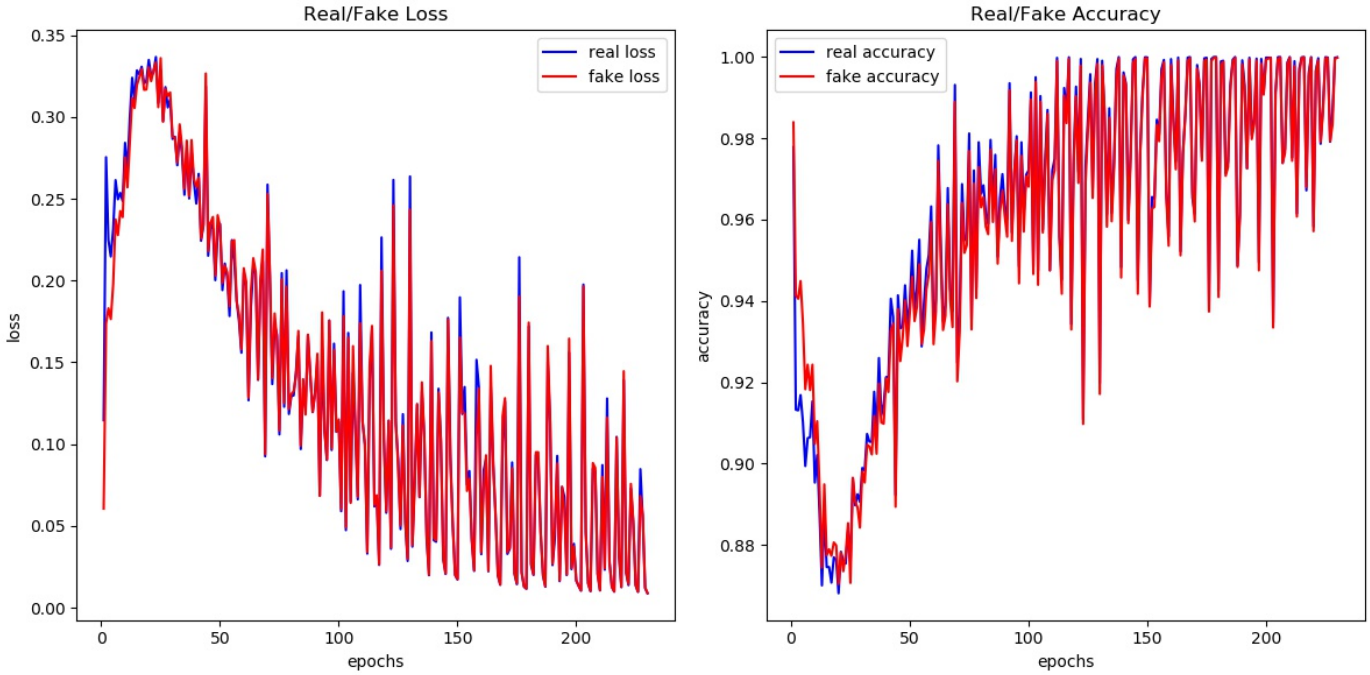


上圖為本次實作的 model，架構上有點像是反過來的 autoencoder。以下更詳細的說明每一層的架構

Discriminator (D)	**Generator (G)**
4 x 4 conv. 128 LeakyReLU. stride 2	4 x 4 conv. 1024 ReLU. stride 2
4 x 4 conv. 256 LeakyReLU. stride 2	4 x 4 conv. 512 ReLU. stride 2
4 x 4 conv. 512 LeakyReLU. stride 2	4 x 4 conv. 256 ReLU. stride 2
4 x 4 conv. 1024 LeakyReLU. stride 2	4 x 4 conv. 128 ReLU. stride 2
4 x 4 conv. 1 Sigmoid. stride 1	4 x 4 conv. 3 Tanh. stride 1

而在 Discriminator 的部分，使用 LeakyReLU，而斜率設為 0.2。而最後一層則是使用 sigmoid，搭配 binary cross entropy 作為 criterion 來計算 output loss。

2-2 Plot the learning curve (in the way you prefer) of your model and briefly explain what you think it represents



左圖為 Discriminator 在判斷影像真偽時，算其 binary cross entropy，real loss 是在給真的影像時，是否判斷為真的影像，而 fake loss 則是在給假的影像時，是否判斷為假的影像。而右圖則是同時算期 accuracy。因此當 binary cross entropy 升高時，同時 accuracy 也會下降。而當 accuracy 越來越接近 100% 時，代表 generator 所生成之圖片足夠可以讓 discriminator 覺得是真的影像。一開始可能 generator 所生成的圖片，discriminator 認為其生成之影像不夠真，在前 50 epochs 的時後有看到明顯往上的趨勢，後來也隨 generator 在 update 參數使得生成影像能越來越貼近真實影像。而 loss 震盪得很厲害，也代表 training 的過程中，互相拮抗的結果，有時會使得 real data 的 output 失真，有時則為 fake data。

2-3 Plot 32 random generated images of your model

2-4 Discuss what you've observed and learned from implementing GAN

可以看到 GAN 所生成的影像，是由 normal distribution 的雜訊通過 generator 所生成的影像，可以看到 GAN 所生成的影像可能會有扭曲的現象，臉部五官看起來並一定在合理的位置上。然而，我們還是可以觀察到生成 quality 很好的影像，如同上圖中第 3 排的影像。這樣的因素可能是因為 GAN 的生成過程比較大局觀，discriminator 同時接受真實影像與假的影像去 update 其參數，可能會學到只要讓 input 越像臉越好，但並一定會得到局部的資，所以局部的地方可能比較不像人臉，如圖中左下角與右下角所生成之圖像。

再者，如果要讓 GAN 運作，discriminator 與 generator 的 capacity 不能差太多，餐數量大概維持差不多，才能使得兩個架構互相影響，互相學習。

在 implement 的時候，有試著將 discriminator update 的次數多於 generator 的次數，像是將 discriminator update 2 次，generator update 1 次，希望能夠使 discriminator 的標準能夠更嚴格，但在這次的實作中結果其實沒有差很多，且同樣的 update 的次數的結果其實也不差，因此最後在實作時使用各 1 次的方式。

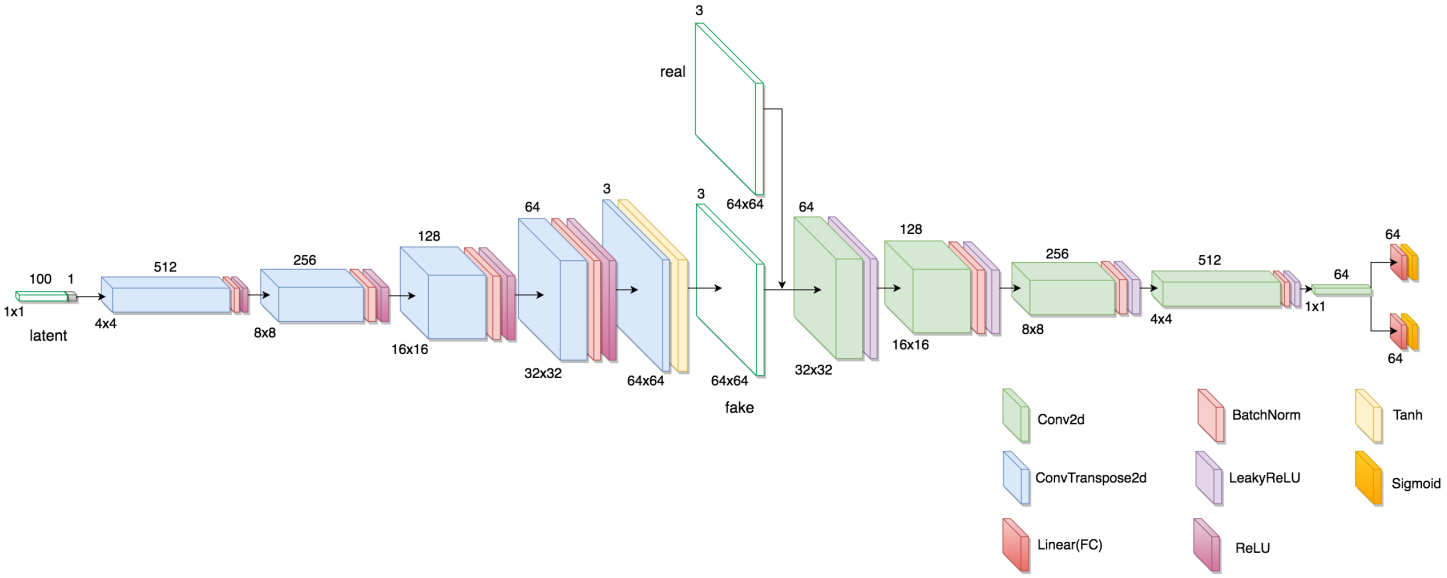
2-5 Compare the difference between image generated by VAE and GAN, discuss what you've observed

可以看到 GAN 所生成的影像，明顯的比 VAE 來得清楚，可能因為 GAN 較有大局觀的效果，因此使得生成的圖片細緻度可以提升，discriminator 認為要越像人臉越好。相較之下 VAE 所生成的影像比較模糊，因為其產生圖片 latent 是由 encoder 出來的，且 output 時比較注重在局部的地方，且在算 loss 的時候要在 reconstruction loss 和 KL divergence 之間做取捨，因此產生的影像才會較為模糊。

Feature Disentanglement

3. Auxiliary Classifier GAN (AC-GAN)

3-1 Describe the architecture and implementation details of your model



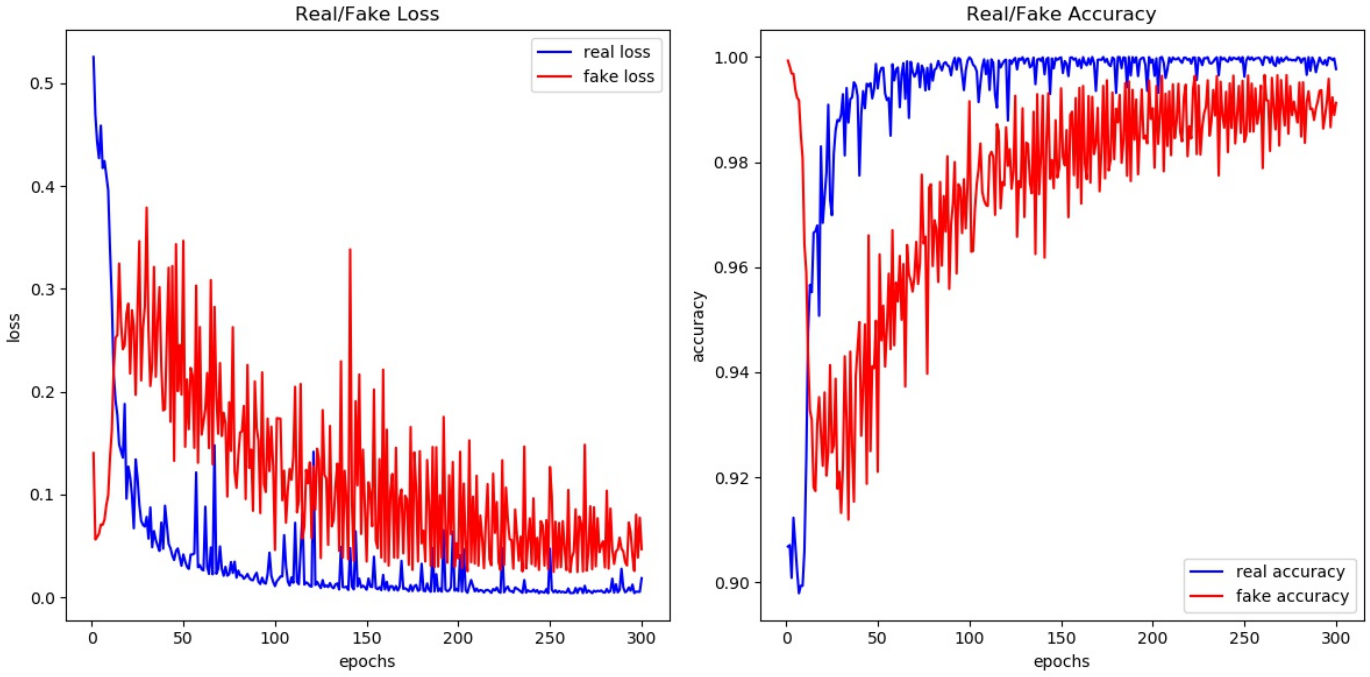
圖中可以看到 latent 多了一個維度，即是要在 AC-GAN 中實作的 smiling attribute。而 output layer 則與前一題 GAN 不同，多了一層 fully connected 的 layer，再搭配 binary cross entropy 作為 classification 與 discriminator 的 criterion。其他的架構則與上一題的設計相同。計算 loss 的時候也是計算 real loss 與 fake loss，不過兩者同時加入 classification 那邊的 loss 一同作 optimize。

另外也有一點不同的是，我們將 discriminator 的第一層的 batch normalization 移除，這是參考 paper 中架構的做法，是希望 discriminator 不要將資訊太快就 normalize 而因此導致失真。

下表則為詳細的 layer 的描述

Discriminator (D) / Recognition (Q)	**Generator (G)**
4 x 4 conv. 128 LeakyReLU. stride 2	4 x 4 conv. 1024 ReLU. stride 2 batchnorm
4 x 4 conv. 256 LeakyReLU. stride 2 batchnorm	4 x 4 conv. 512 ReLU. stride 2 batchnorm
4 x 4 conv. 512 LeakyReLU. stride 2 batchnorm	4 x 4 conv. 256 ReLU. stride 2 batchnorm
4 x 4 conv. 1024 LeakyReLU. stride 2 batchnorm	4 x 4 conv. 128 ReLU. stride 2 batchnorm
4 x 4 conv. 64 stride 1 FC. output layer Sigmoid for D FC. output layer Sigmoid for Q	4 x 4 conv. 3 Tanh. stride 1

3-2 Plot the learning curve (in the way you prefer) of your model and briefly explain what you think it represent



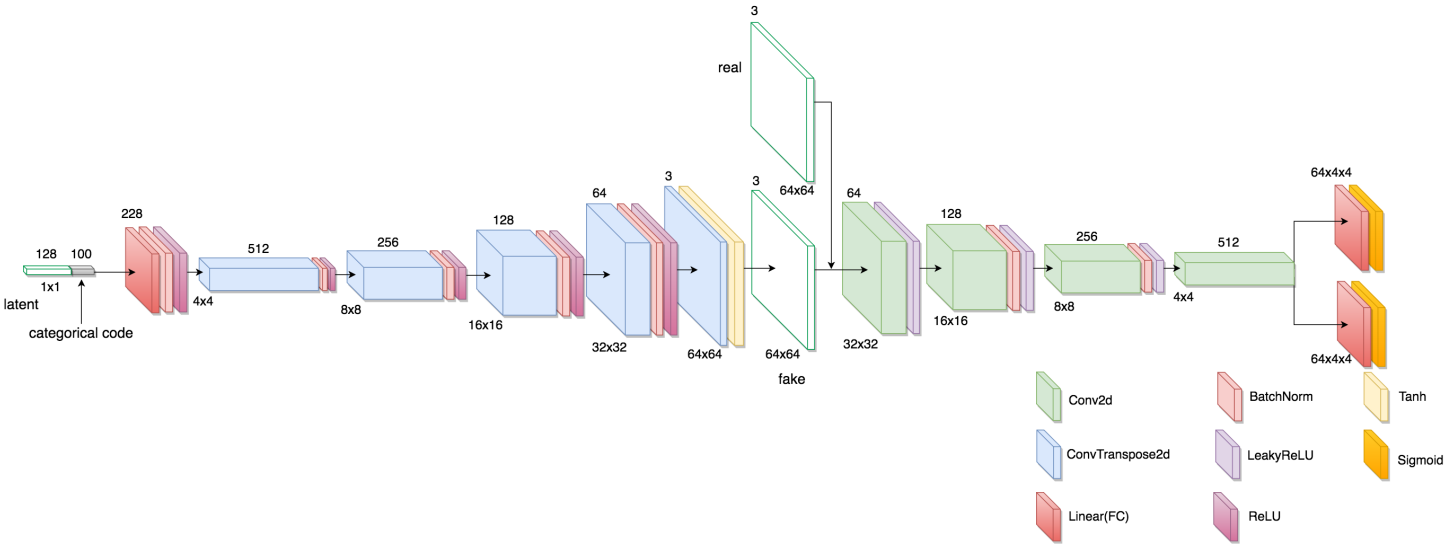
同樣的可以看到當 loss 上升時，相對應的 accuracy 也是下降的。而在 implement AC-GAN 當中，我們將 discriminator 的參數先 update 兩次，再 update generator 一次。因此，discriminator 學習的速度會比較快，可以看到 real loss 的部分，明顯的看到下降的速度很快，且很快地接近 0，real accuracy 因此也很快上升到 1。這代表著 discriminator 有較強的能力認出真實的影像，real loss 下降的同時，可以看到 fake loss 同時上升較大的幅度，可見在這段時間 discriminator 認為 generator 生成的影像品質不夠好，可見 discriminator 對影像的分別的能力足以分辨真偽。且後來也有 fake loss 相對有比較大的跳動，可能是因為在實作中將 fake attribute 的 loss 也算入，隨機給的 label 讓他的 loss 跳動較為劇烈。

3-3 Plot 10 pair of random generated images of your model, each pair generated from the same random vector input but with different attribute. This is to demonstrate your model's ability to disentangle feature of interest

在實作時使用 smiling 這個 attribute，latent 為 100 再加上一個 attribute 的維度，總共 101 維。上圖中，在上排是 attribute 為 0 的狀況，而下排的則為有包含此 attribute (1)。較明顯的就是讓嘴吧的部分加上牙齒，看起來就像在笑。但有的則是嘴角微微上揚，實作上也不太清楚為何 model 可以學到這樣的結果。另外在實作中，前幾個 epoch 中生成的影像較差，比較像是雜訊時，似乎也是有看到嘴吧的部分生成的結果也有略為不同的情況。

4. Information Maximizing GAN (Info-GAN)

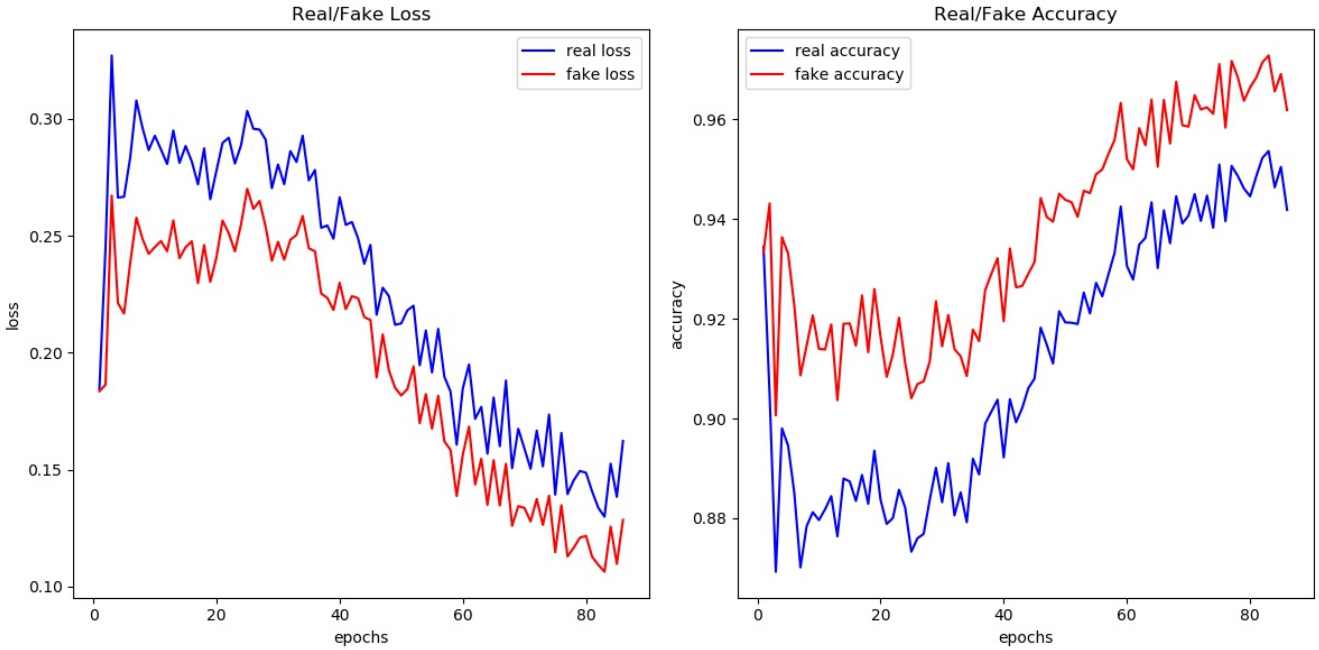
4-1 Describe the architecture and implementation details of your model



架構上做了一些更動，generator 前面加上一層 fully connected layer，discriminator 部分則將最後一個 convolution layer 併到 fully connected layer 內。以下為詳細的 model 架構

Discriminator (D) / Recognition (Q)	**Generator (G)**
	FC. 448 x 2 x 2 output layer Sigmoid for D
4 x 4 conv. 64 LeakyReLU. stride 2	4 x 4 conv. 1024 ReLU. stride 2 batchnorm
4 x 4 conv. 128 LeakyReLU. stride 2 batchnorm	4 x 4 conv. 512 ReLU. stride 2 batchnorm
4 x 4 conv. 256 LeakyReLU. stride 2 batchnorm	4 x 4 conv. 256 ReLU. stride 2 batchnorm
4 x 4 conv. 512 LeakyReLU. stride 2 batchnorm	4 x 4 conv. 128 ReLU. stride 2 batchnorm
FC. output layer Sigmoid for D FC. output layer Sigmoid for Q	4 x 4 conv. 3 Tanh. stride 1

4-2 Plot the learning curve (in the way you prefer) of your model and briefly explain what you think it represents



Discriminator 中的 activation function: LeakyReLU 的斜率，在這裡使用 0.1，也與 paper 中的 model 相同。

在這個實作中，recognition network (Q) 那邊選用與原始 paper[2]相同的架構，使用 10 ten-dimensional categorical code 來作為 recognition network 的 output，因此 latent space 的維度為 128 維的 random noise 加上 100 維的 categorical code。而實作中發現，discriminator 在辨別真偽的部分 update 其實較為容易，但是 recognition network 的部分其實較難讓其 loss 下降，反而持續持平或略為上升的現象。加上此架構為 unsupervised 的架構，從 random label 學習到影像中的特徵，其實實作上略為困難，可以看到 loss 相較其他而言來得較高，就 real loss 而言，在 AC-GAN 中是 discriminator 的 loss 和 recognition network 的 loss 相加，而 infoGAN 則只有計算在 discriminator 的 loss，相比之下就可以知道 supervised 與 unsupervised 的差異。而在原始 paper 中則是利用 information theory 來說明 infoGAN 的架構可以成功，且做了不同的實驗。但在給定隨機的 label，我們也無法知道 model 特定的 vector 學到什麼樣的 attribute，因此實作上 train 了幾次，attribute 變化的結果都有些不同。

4-3 Plot random generated images of your model

上述有提到每次的結果都有略為不同，以下則是分別兩次 train 的 model 得到的結果

每張圖都是固定 128 維的 random noise，改變 category 的 label。可以看到這裡差別在於髮色的不同。

而相同的 model，實驗中得到不同的結果，這個則是 smile 的樣子的不同，可以看到有些笑起來較為燦爛，有的像是左 5 則只有微微的笑。

4-4 Discussion

InfoGAN 在 training 上很難 train，在實作上其實很容易結果出來全部都是雜訊，原本以為 train 不起來，沒想到多試了幾次就有結果。之後就開始慢慢的感覺有臉部特徵的感覺。目前這個 model 是使用 paper 上提到 10 個 ten-dimensional categorical code，才有這樣的結果。另外也有試驗 1 個 ten-dimensional categorical code，但結果出來，每個 category 的影像相似度很高，沒有分別的感覺。

之前想過利用前幾個 epoch 利用 supervised 的方式給 model 一點資訊，過幾個 epoch 之後再使用 random 的 label 給 model，但因為時間的因素而沒有實作。

---

[2] InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, Chen et al., NIPS 2016 https://arxiv.org/pdf/1606.03657.pdf