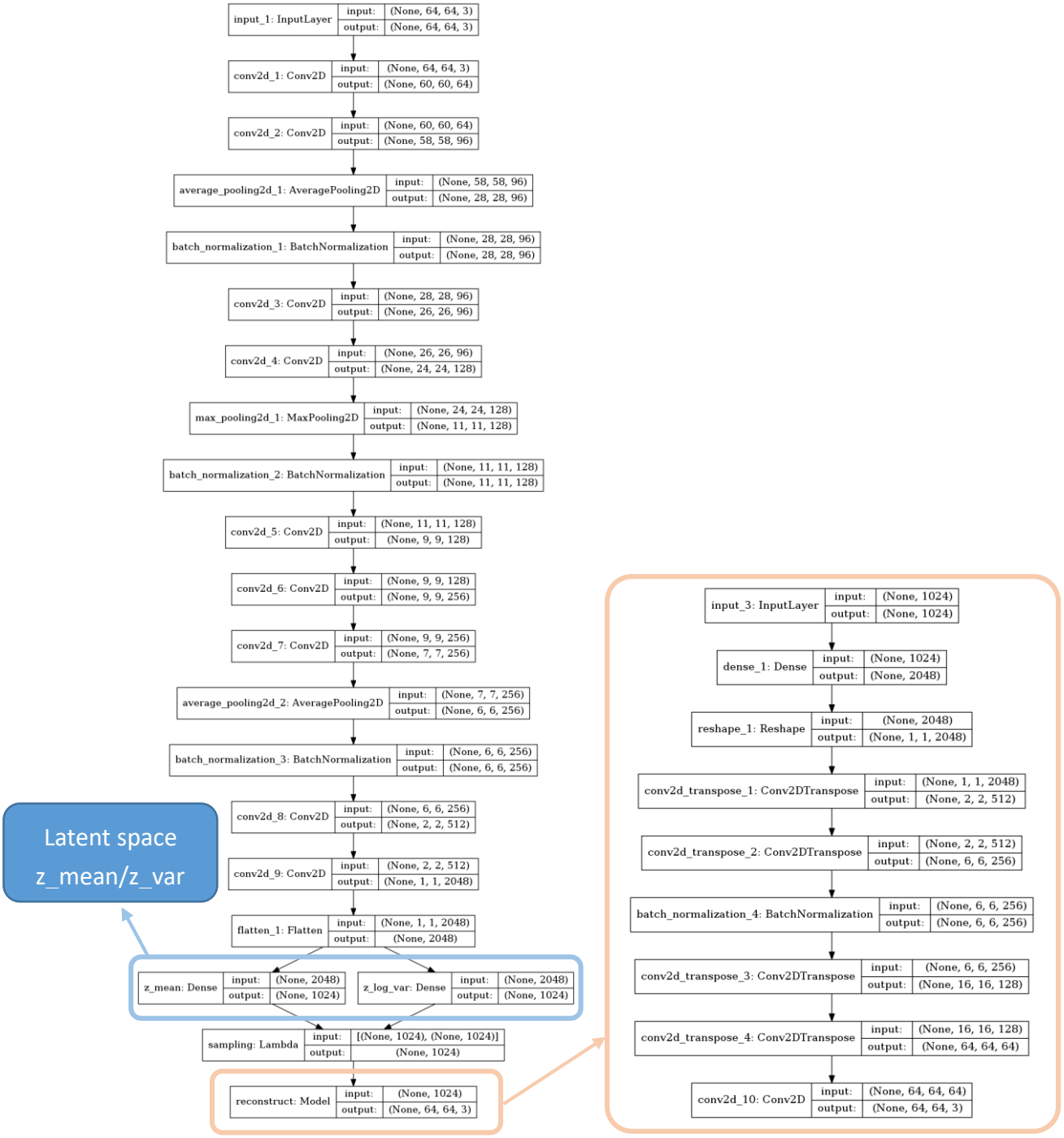
HW4

r06921062 蘇楷鈞

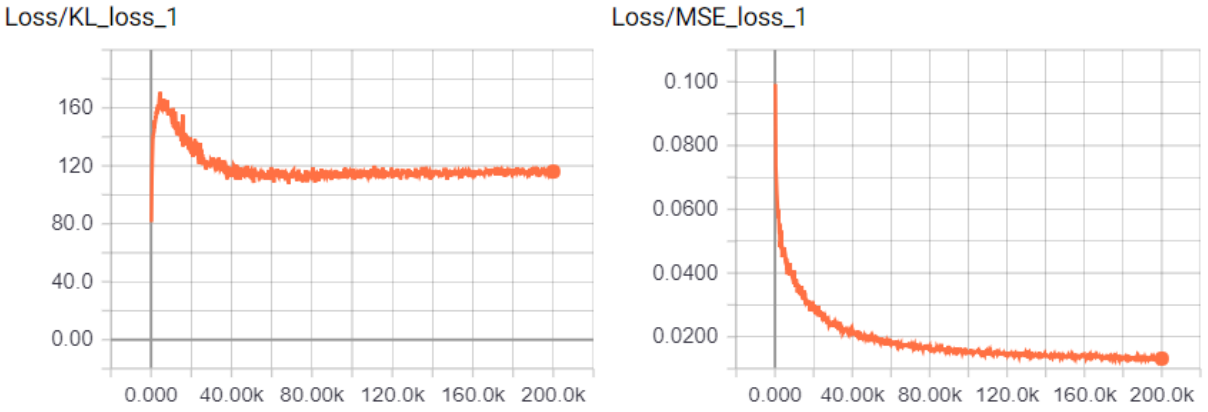
Problem 1. VAE

[1] Describe the architecture & implementation details of your model. (1%)



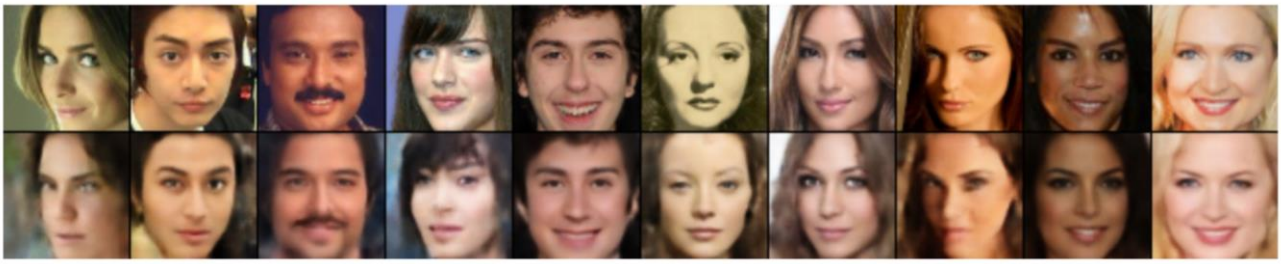
Loss function	Optimizer	Batch size	Epoch	Latent dimension
MSE+ 5e-5*KL_loss	Adam(2e-4) Decay=0.9step=4000	200	500	1024

[2] Plot the learning curve (reconstruction loss & KL divergence) of your model [fig1_2]. (1%)



[3] Plot 10 testing images and their reconstructed result of your model [fig1_3] and report your testing MSE of entire test set. (1%)

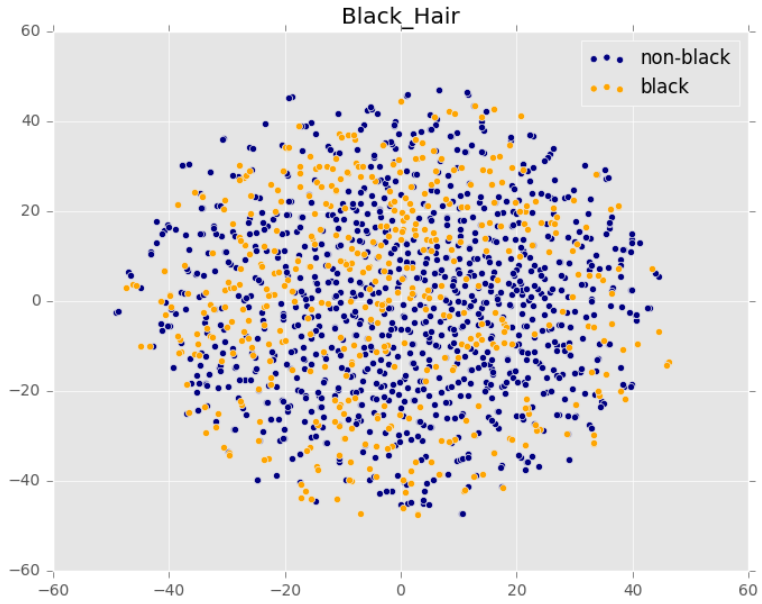
上方為 test data 的原圖，下方為重建後的圖片。



MSE	0.958364

[4] Plot 32 random generated images of your model [fig1_4]. (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 [fig1_5]. (1%)



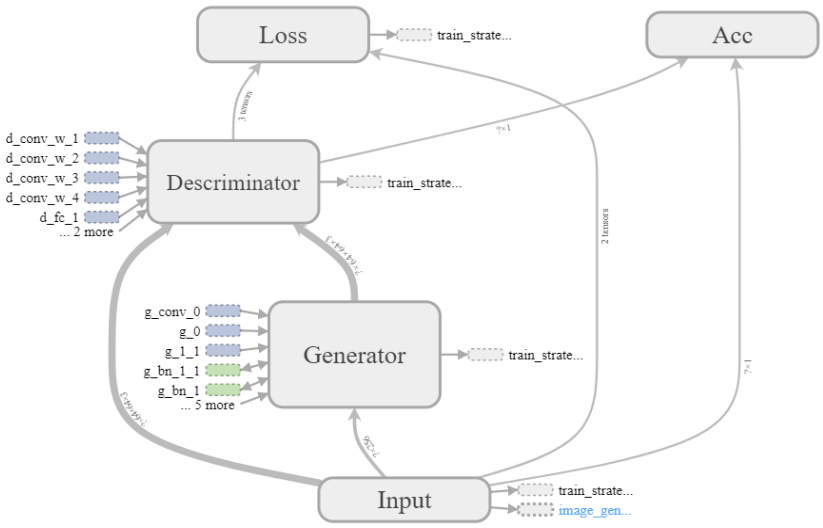
[6] Discuss what you've observed and learned from implementing VAE. (1%)

在實作 VAE 前，有先試試 AutoEncoder，在訓練時發現 latent dimention 得設 512 才有比較好的還原效果；而在 VAE 實作時，同樣維度生成的圖片較糊，得設為 1024 時才有較好的效果。猜測是因為 VAE 是從機率分佈找出點來還原，而 AE 只需把壓縮後的資訊還原，因此兩者目的不同，所以要在機率分佈中塞入更多資訊，只好擴大維度，然而也因為是從機率分佈中還原資訊，因此在還原隨機向量時有較 AE 更好的效果。

Problem 2. GAN

[1] Describe the architecture & implementation details of your model. (1%)

- Model architecture



- Model hyperparameter

<table>
<tr><th></th><th colspan="2">Layer name (kernel size) / Filters / strides</th></tr>
<tr><td rowspan="8">Generator</td><td colspan="2">Conv2D (1x1) / 4096 / 1</td></tr>
<tr><td colspan="2">Reshape(2,2,1024)</td></tr>
<tr><td colspan="2">Conv2D_Transpose (4x4) / 512 / 2</td></tr>
<tr><td>BatchNormalization</td><td rowspan="3">Conv2D_Transpose (16x16)/
128 / 8</td></tr>
<tr><td>Conv2D_Transpose (4x4)/ 256 / 2</td></tr>
<tr><td>Conv2D_Transpose (9x9) / 128 / 4</td></tr>
<tr><td>BatchNormalization</td><td>BatchNormalization</td></tr>
<tr><td colspan="2">Add</td></tr>
<tr><td colspan="2">Conv2D_Transpose (5x5) / 64 / 2</td></tr>
<tr><td colspan="2">Conv2D (1x1) / 3 / 1</td></tr>
<tr><td colspan="2">tanh</td></tr>
</table>

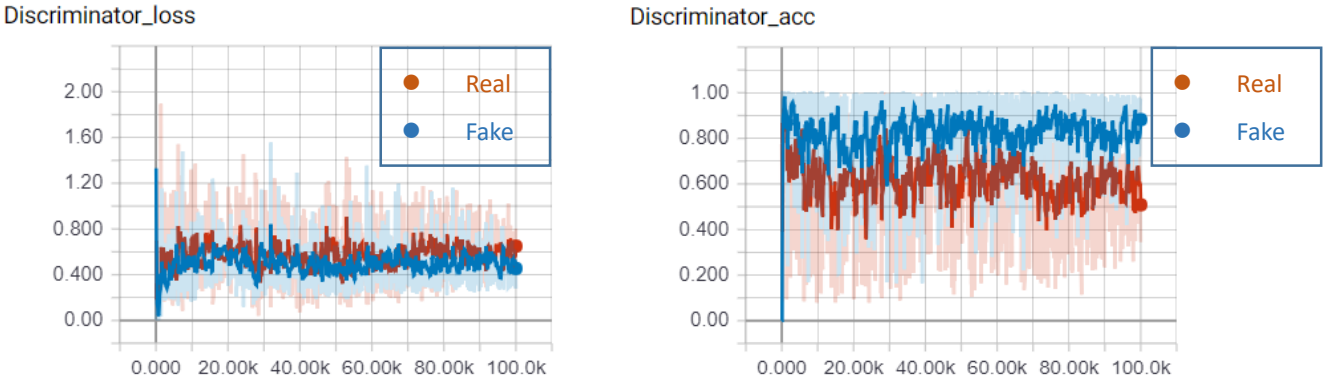
<table>
<tr><th></th><th>Layer name (kernel size) / Filters / strides</th></tr>
<tr><td rowspan="9">Descriminator</td><td>Conv2D (5x5) / 256 / 2</td></tr>
<tr><td>Conv2D (3x3) / 128 / 1</td></tr>
<tr><td>avg_pool2d (3,3) / - / 2</td></tr>
<tr><td>Conv2D (3x3) / 64 / 1</td></tr>
<tr><td>Conv2D (3x3) / 64 / 2</td></tr>
<tr><td>flatten</td></tr>
<tr><td>Fully-connected / 256</td></tr>
<tr><td>Fully-connected / 64</td></tr>
<tr><td>Fully-connected / 1 / sigmoid</td></tr>
</table>

- Training method

Optimizer	Batch size	Iteration	Latent dimension	Input domain
Descriminator : RMSProp(1e-4) Generator : Adam(5e-5)	200	100000	256	[-1,1]

[2] Plot the learning curve (in the way you prefer) of your model and briefly explain what you think it represents [fig2_2]. (1%)

GAN 在訓練過程十分不穩定，fake 和 real 的 loss 會一高一低來回震盪，通常會是第 t 個 iter 時 real>fake，t+1 個 iter 會變成 fake>real，反之亦然，代表 generator 會在 discriminator 較強時，學到較多資訊，導致下一個 iter 時，discriminator 無法馬上更新出新的辨別規則。另外，loss 維持在 0.6 附近應該是較理想的值，雖然以準確率來看，應該有較強的能力分辨出假的圖片，但依數學推論得到平均答對的機率會在 0.5 到 0.6，在更新時也應該有不錯的效果。



[3] Plot 32 random generated images of your model [fig2_3]. (1%)

[4] Discuss what you've observed and learned from implementing GAN. (1%)

在訓練 GAN 時，將 pixel domain 限制在[-1,1]效果較佳；調整 discriminator 的強度十分重要，一開始常讓 discriminator 完全分辨出真假，使得 generator 產生不出好圖片，藉由調整參數量、lr、Optimizer，使得 loss 變化再 0.6 附近，從 cross entropy 來看，0.6 大約就是 discriminator 預測的機率平均在 0.5 附近，藉此訓練出的效果較佳。

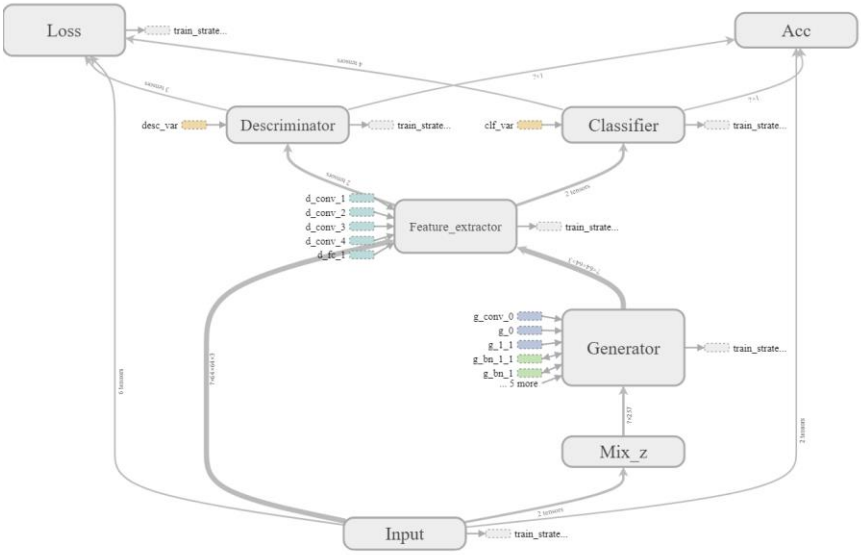
[5] Compare the difference between image generated by VAE and GAN, discuss what you have observed. (1%)

從 VAE 到 GAN，可以發現 VAE 因為 loss 裡有 pixel-wise 的 MSE，使得 generator 學到的是盡力還原成和真實圖片一樣，使得對隨機向量產生的結果會有一些奇怪的區塊且較為模糊；而 GAN 較能把隨機向量產生較為完整且較清晰的圖片。然而在 GAN 的結果上，邊緣時常有不明的亮點，目前還無法理解原因。

Problem 3. ACGAN

[1] Describe the architecture & implementation details of your model. (1%)

- Model architecture



- Model hyperparameter

	Layer name (kernel size) / Filters / strides
Generator	Conv2D (1x1) / 4096 / 1
	Reshape(2, 2, 1024)

	Conv2D_Transpose (4x4) / 512 / 2		
	BatchNormalization	Conv2D_Transpose (16x16)/ 128 / 8	
	Conv2D_Transpose (4x4)/ 256 / 2		
	Conv2D_Transpose (9x9) / 128 / 4		
	BatchNormalization	BatchNormalization	
	Add		
	Conv2D_Transpose (5x5) / 64 / 2		
	Conv2D (1x1) / 3 / 1		
	tanh		

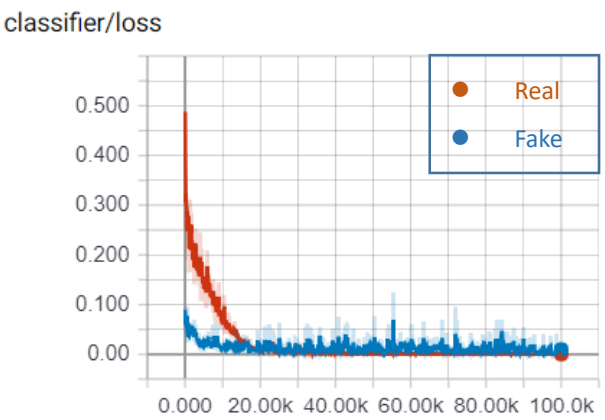
		Layer name (kernel size) / Filters / strides	
Extractor		Conv2D (5x5) / 256 / 2	
		Conv2D (3x3) / 128 / 1	
		avg_pool2d (3,3) / - / 2	
		Conv2D (3x3) / 64 / 1	
		Conv2D (3x3) / 64 / 2	
		flatten	
		Fully-connected / 256	
Desc	Clf	Fully-connected / 64	Fully-connected / 64
		Fully-connected / 1 / sigmoid	Fully-connected / 1 / sigmoid

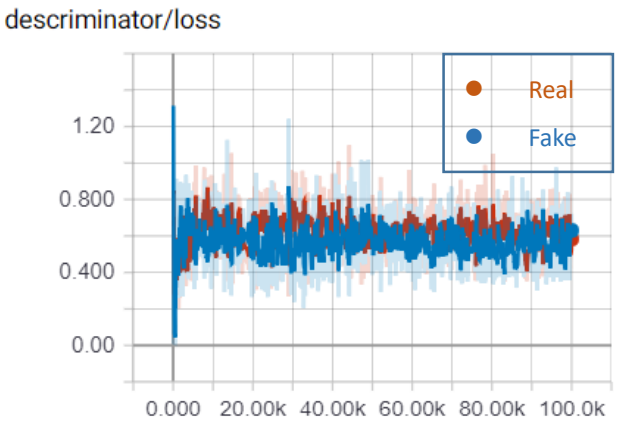
- Training method

Optimizer	Batch size	Iteration	Latent dimension	Input domain
Descriminator : RMSProp(5e-5) Classifier : Adam(1e-4) Generator : Adam(1e-4)	200	100000	256	[-1,1]

[2] Plot the learning curve (in the way you prefer) of your model and briefly explain what you think it represents [fig3_2]. (1%)

由於 classifier 是同時訓練的，因此一開始並無法做準確的分類，然而 generator 初期很容易被分類正確，應該是 generator 本身在更新時就往 classfier 最大化的方向前進，容易抓到目前 classifier 是依據什麼來分類，使得產生的所有圖片都往那特徵走，變異較小，而在 real 的圖片上則因為變異較大，因此需要一些訓練時間才能正確分類，所以 generator 在較後期學到的才會是較正確的特徵。而 discriminator 則和一般的 GAN 一樣。





[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. (2%)

可看出 generator 學到的笑臉是嘴巴上揚且顴骨周圍會較不笑的圖片明顯上升。

Bonus InfoGAN

[1] Describe the architecture & implementation details of your model.

同 ACGAN，差別在 classifier 換成 7 個 class 且 activation 改為 softmax，且沒有使用真實圖片的類別。

Loss function	Optimizer	Batch size	Iteration	Latent dimension
Classifier : Categorical crossentropy	Descriminator : RMSProp(5e-5) Classifier : Adam(1e-4) Generator : Adam(1e-4)	200	70000	256

[2] Plot 5 random generated images with 7 different categories and discuss feature of each category.

笑臉
臉變瘦
男性
替嘴唇上妝
向左看
女性
向右看

[3] Discuss model problem.

在訓練 InfoGAN 時，一開始採用 word embedding 的方法，容易 mode collision，可能原因為 generator 為了讓 classifier 能正確分類，走向讓 random noise 的權重變為 0，讓 class 的資訊直接傳遞到圖片，因此後續皆採用 one hot 的方式。在不同 class 上特定類別在某些 random noise 上會做不好，例如在向左看的圖片中，第一張可能本身就帶有些向右看的資訊，使得他往左看時出錯，而第二張本身就有些向左看，在加了這類別後，臉有些變形了。另外，設計上是用 one hot 的方式，但實際上還是能嘗試混和兩個 label 以產生圖片，例如下圖為笑臉和向右看的結合。