深度學習結合智慧家庭監控

0410009 羅宇呈 0410024 余東儒

研究動機

- 近年來，深度學習已然掀起一波科技風潮，至今仍不斷的發展茁壯
- 伴隨著電腦的運算能力增強，以及大數據的時代，讓模擬神經網路變的可能
- 深度學習並不受限於領域，相反的，它更像是工具，學會使用它，必能在研究或應用上，產生不小的助益。
- 因此，我們便打算以深度學習作為專題題目，並試著應用深度學習來解決實際的問題。

研究題目

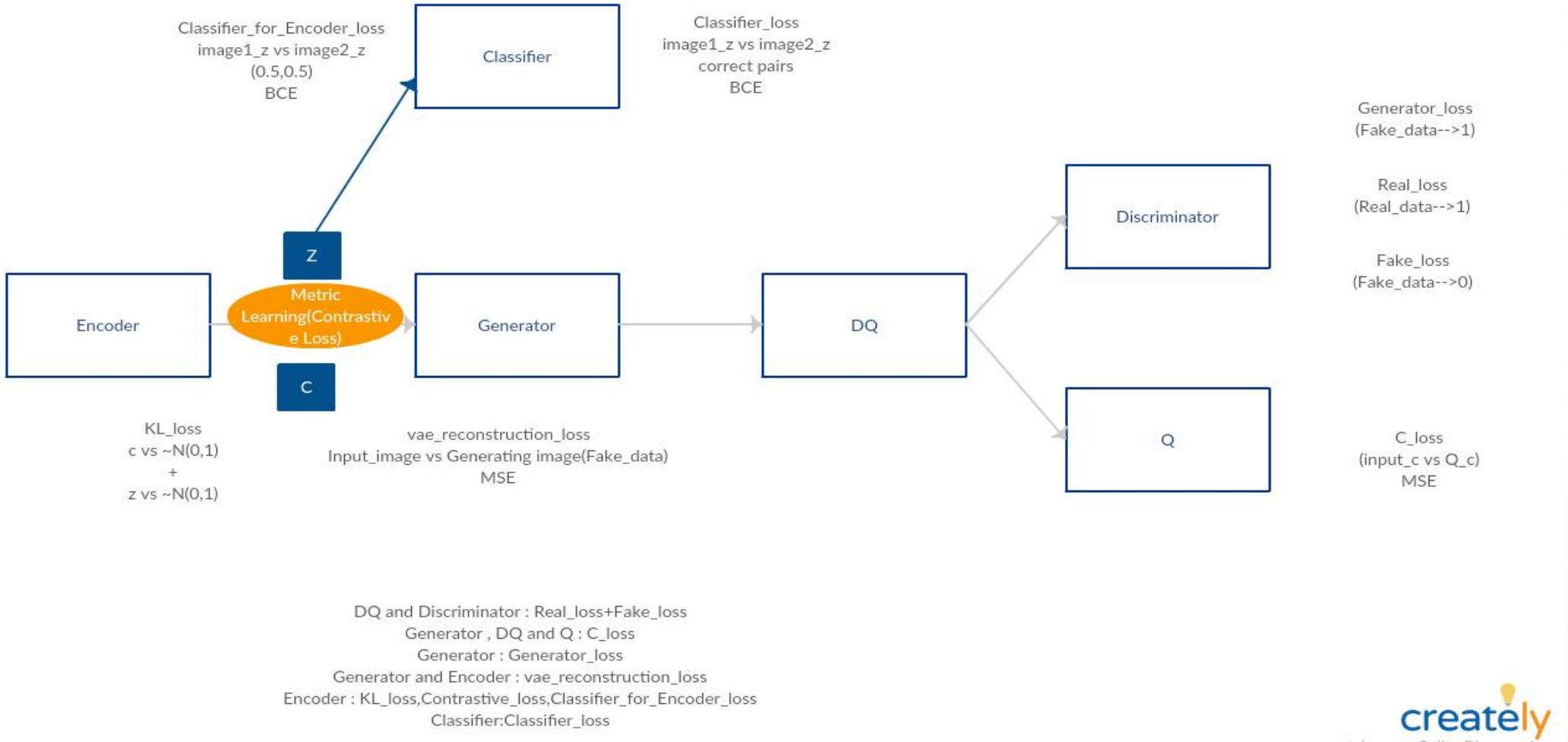
- 人臉辨識是深度學習的一個應用實例，應用在手機、平板、機場海關等地方，現已不足為奇

- 我們想到，結合智慧家庭的概念，可將深度學習的概念，應用於家庭監視錄影機，
  使其能”自動”辨識家庭成員，同時，因為監控的人數較少，我們希望我們設計出的演算法，能夠有以下幾個特點:

- 相較於大量的人臉集合辨識，我們著重的是家庭監控，
  因此我們希望提升在相對封閉環境(獲少量集合)時的辨識能力、甚至辨識速度。

- 我們希望訓練出的模型具有主動式學習的特性，也就是適應變化的能力，
  並在訓練以後，能夠成功將不同的人分群，或是因應成員的加入而自行產生新的資料群

專題架構圖


Classifier_for_Encoder_loss
image1_z vs image2_z
(0.5,0.5)
BCE
Classifier
Classifier_loss
image1_z vs image2_z
correct pairs
BCE
Generator_loss
(Fake_data-->1)
Real_loss
(Real_data-->1)
Fake_loss
(Fake_data-->0)
Discriminator
z
Encoder
Metric Learning(Contrastive Loss)
Generator
DQ
c
Q
KL_loss
c vs ~N(0,1)
+
z vs ~N(0,1)
vae_reconstruction_loss
Input_image vs Generating image(Fake_data)
MSE
C_loss
(input_c vs Q_c)
MSE
DQ and Discriminator : Real_loss+Fake_loss
Generator , DQ and Q : C_loss
Generator : Generator_loss
Generator and Encoder : vae_reconstruction_loss
Encoder : KL_loss,Contrastive_loss,Classifier_for_Encoder_loss
Classifier:Classifier_loss
creately
www.creately.com • Online Diagramming

研究過程

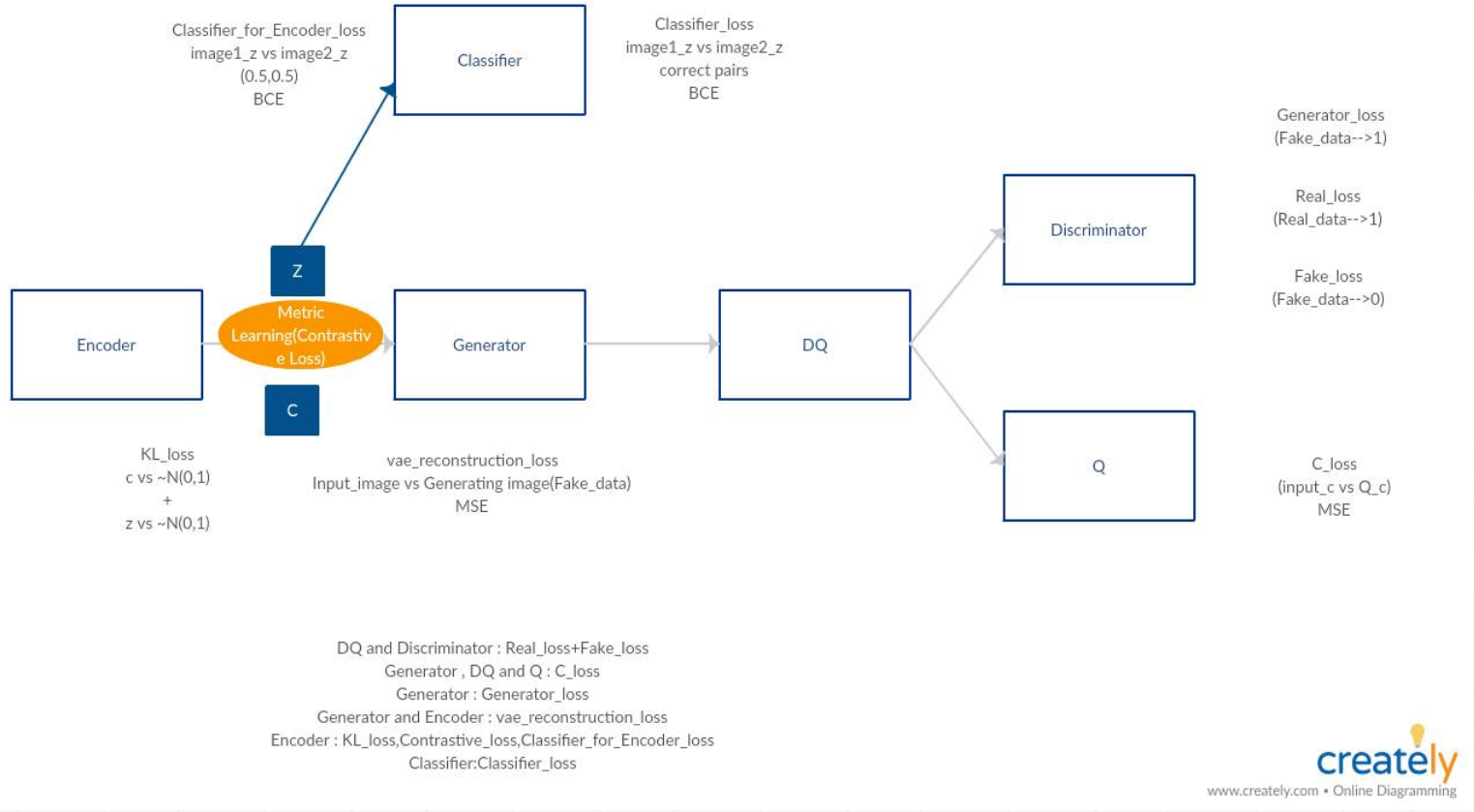
- 以下將一步一步介紹架構:

  (I)對input進行處理

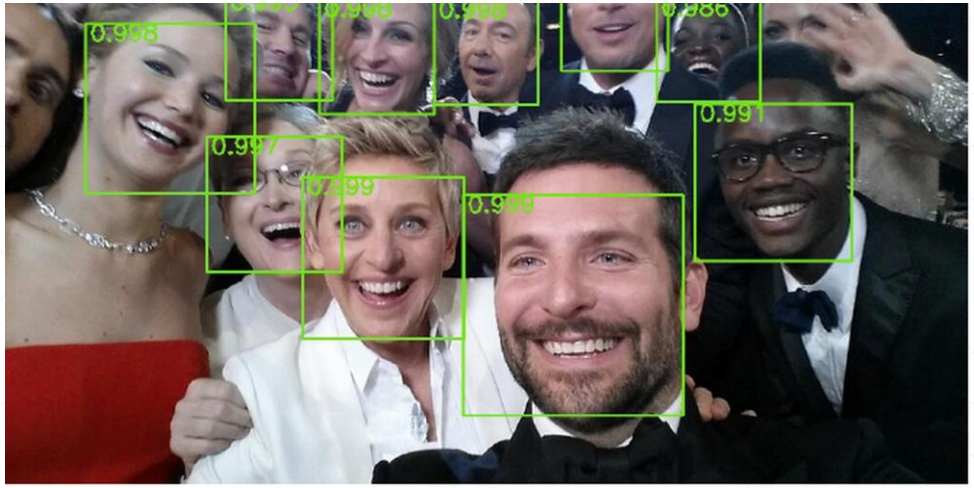
  (II)訓練Encoder、Classifier

  (III)訓練infoGAN

  (IV)整體訓練過程

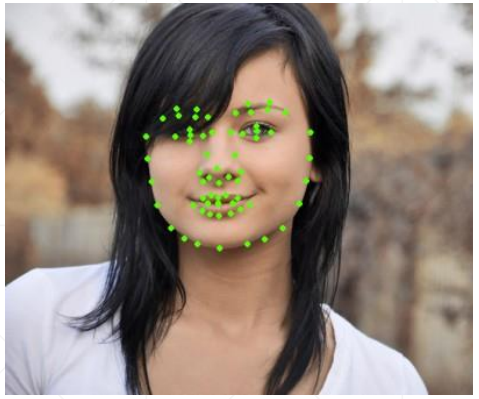


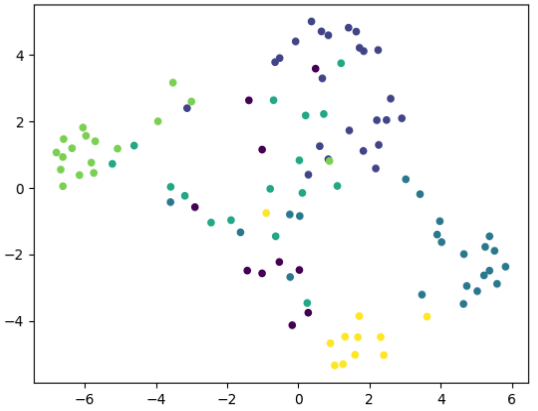
(I)對input進行處理



- Input : video(攝影機)
- Video -> frame
- 利用face detection的方式，找出frame中所有的人臉並切下，同時記錄這些人臉所屬的frame
- 至此，我們建立一個由人臉組成的dataset，並擁有其frame資訊。

(II)訓練Encoder、Classifier





- Encoder:從人臉中分離出帶有ID(Identity)的資訊

將input的人臉圖片，encode成face feature並投射在latent space上，並利用disentanglement的 方式，將encode的結果變成一個C+Z的vector

C代表我們要的ID資訊，Z則代表不包含ID的其他資訊，此處，我們設C=1維，Z=19維

- 我們期望C+Z在latent space會有根據C cluster的現象->利用Metric Learning的想法，把相同C的兩張圖片稱為positive pair，反之則稱為negative pairs，利用contrastive loss，讓positive pairs更靠近，negative pairs更遠，以達到cluster的效果。

Classifier

(1,0)

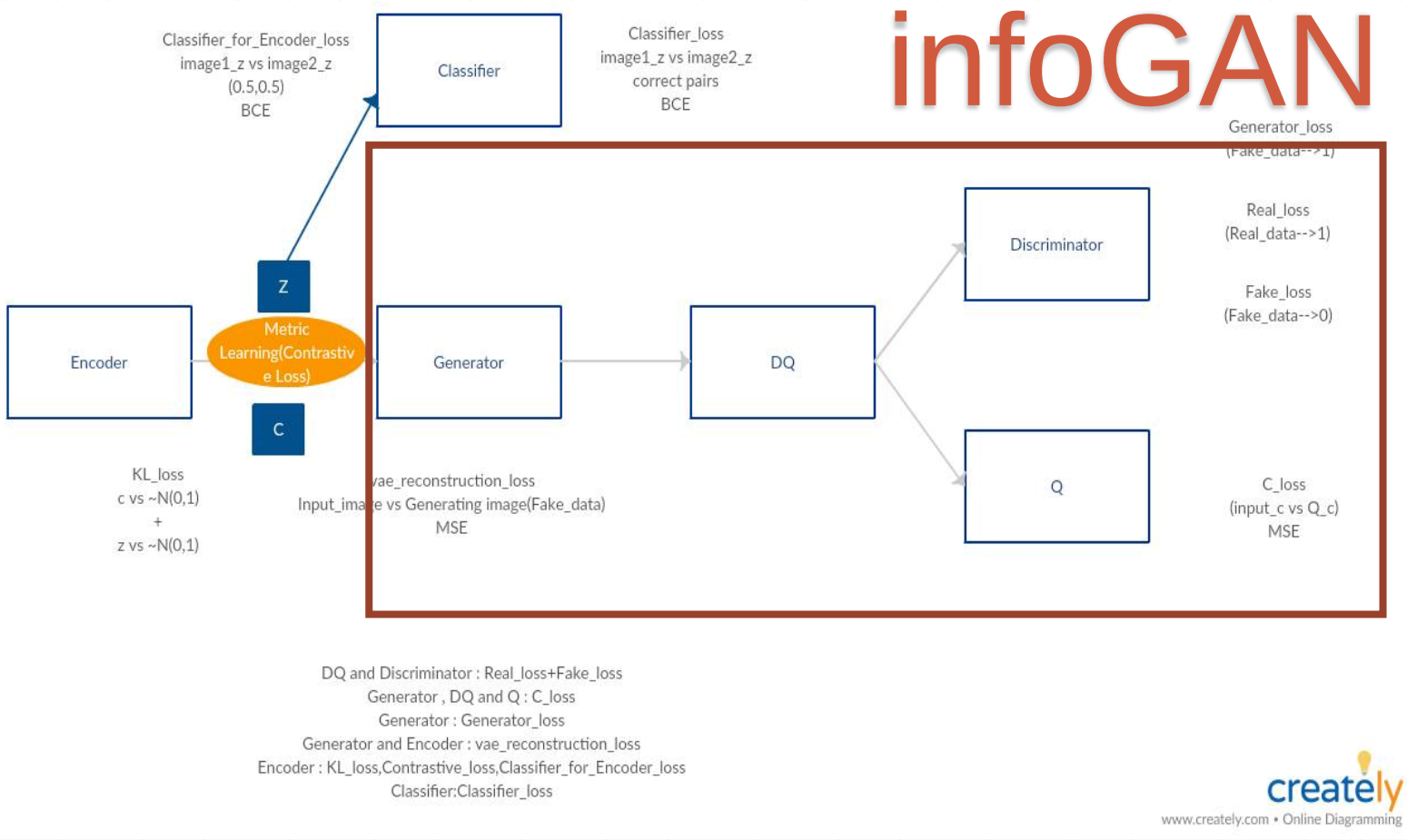
(1,0)

(0,1)

- 顧名思義，為辨別input的兩張圖片是否為同一個人(C相同)
- 用來與Encoder達到對抗(adversarial)的效果。
- Classifier的Input為兩張圖片的Z，因為Z不含ID資訊，因此理論上，當Classifier接收到Input時，應該要無法辨別
- Encoder期望能夠將圖片的ID資訊分離出來成C，其餘為Z；Classifier則期望從兩張圖片的Z中辨別圖片，因而形成相互對抗的效果，因此此處有兩個loss : Classifier_Loss用來update Classifier使其辨別能力更強，Classifer_for_Encoder_Loss則用來讓Encoder學習如何分離C與Z。

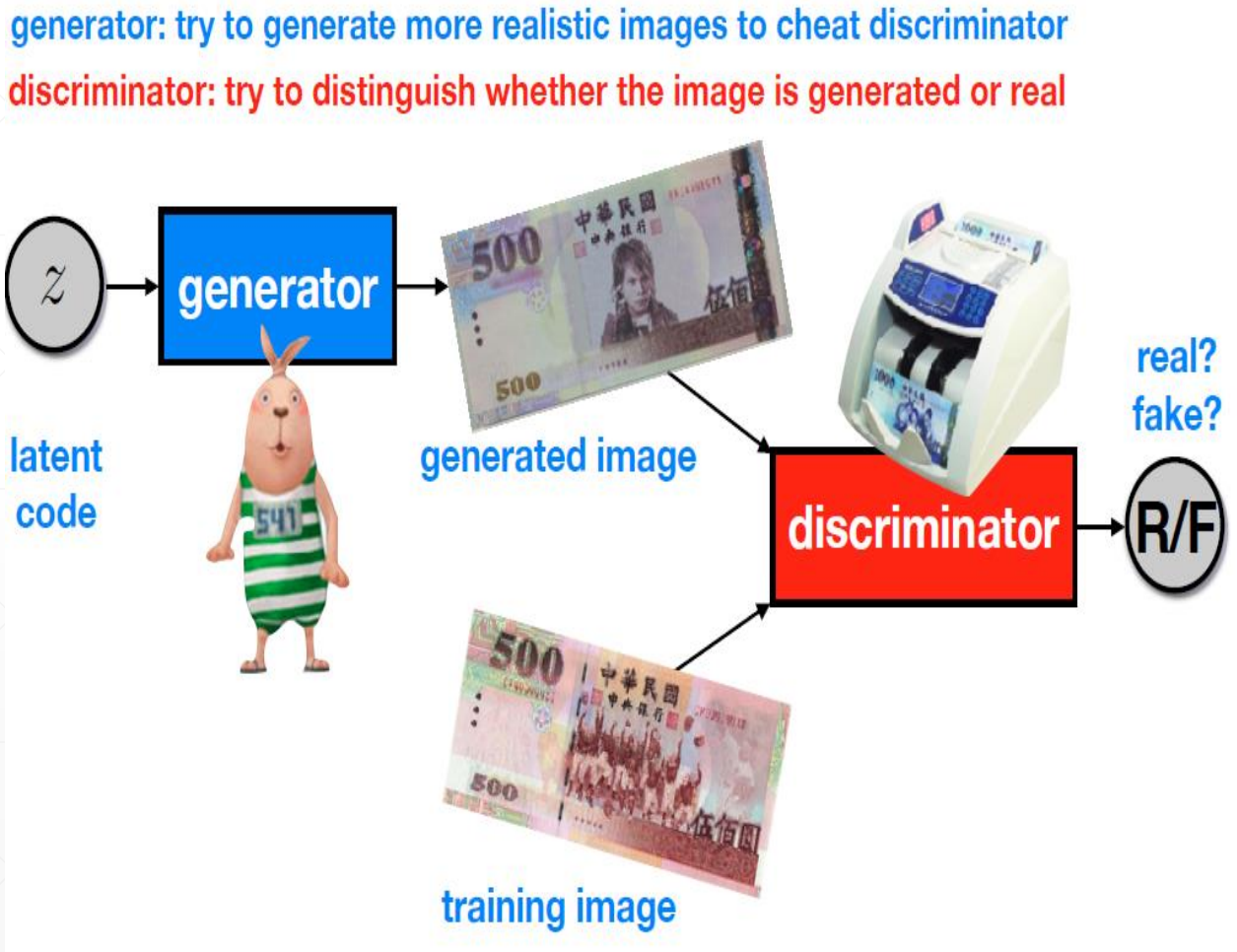
(III)訓練infoGAN

- What is infoGAN?

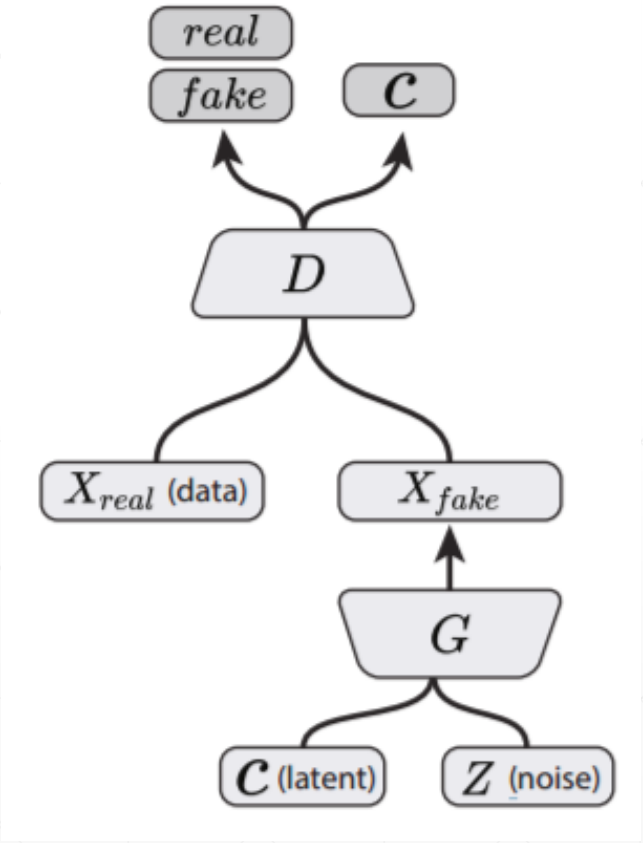


Generative Adversarial Network (GAN)

- Generator V.S Discriminator
- 製作假鈔的機器 驗鈔機
- 畫假畫謀生的人 鑑定官
- 以假亂真 明察秋毫



InfoGAN



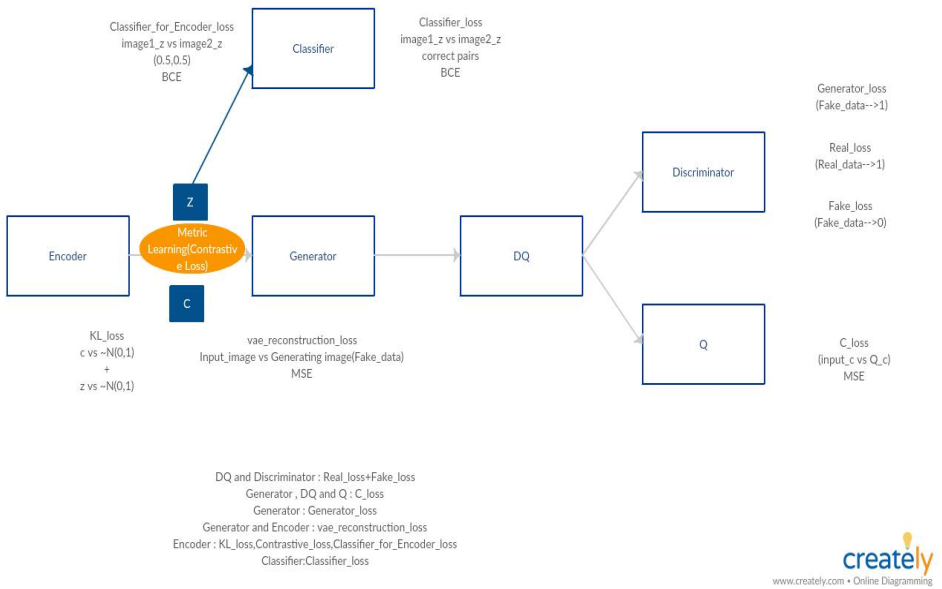
- Generator的input有兩種:
  1.E(x)，即來自encoder的結果
  2.random sample from Z、C

- Generator把input 重新建造(reconstruct)成image，並往後送往Discriminator與Q，
  架構圖中DQ為Discriminator與Q的common layer。
  此處的Loss為reconstruction_loss
  (比較real_image以及reconstructed_image)。

Discriminator

- input共有兩個來源:

1. 額外給的real image
2. 2.Generator reconstruct的image(稱為fake image)



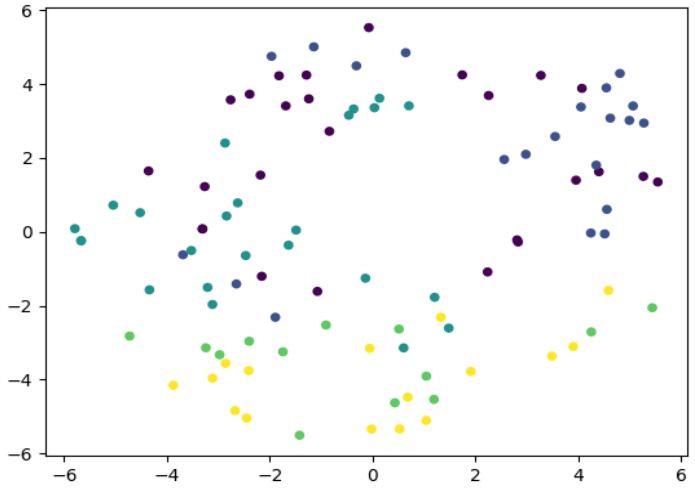
- Discriminator試圖辨別input的圖片是否為真，true-> 1，fake-> 0
- 同時因為GAN的特性，Generator與Discriminator將互相競爭
- Generator試圖騙過Discriminator，而Discriminator則逐漸提升自己的辨識力，因此此處應有3個Loss:Fake_Loss(Fake->0)、Real_Loss(Real->1)、Generator_Loss(Fake->1)，前兩者用來update Discriminator、DQ，後者則用來update Generator。

Q

Homer!

- Q的部分則是InfoGAN “Info”的部分。
- reconstructed image 經過DQ common layer後同樣會送至此處，此處Q將會猜測Generator是根據甚麼C來reconstruct這張image的
- 因此此處的loss為C_Loss(比較Q所猜測的C以及實際上real image的C)，並用來update DQ、Q以及Generator。

(IV)整體訓練過程



- 目前的訓練模式為:

  前20個Epoch專門用來訓練Encoder與Classifier(階段(II))

  後50個Epoch則將後面的InfoGAN架構加入一起訓練(階段(II)+階段(III))
- 我們選擇的batch size為固定100，由於每個Dataset的大小不同，iteration的數量也不同
- 在每個iteration中，Generator根據兩種不同input(E(X)，random sample from Z、C)而update 的次數分別為1:1

  之後再用Positive/Negative Pairs update Encoder和Classifier 2次
- 這是因為我們從最後的t-SNE圖發現，若只update Encoder和Classifier 1次，分群的效果並不好，

  因此選擇update 2次，才有比較漂亮的結果。

研究結果

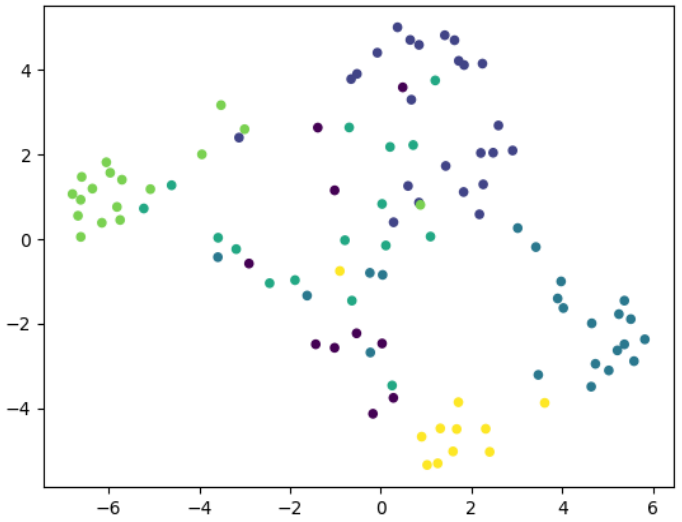
- (I)T-ARA - DAY BY DAY

網址如下:

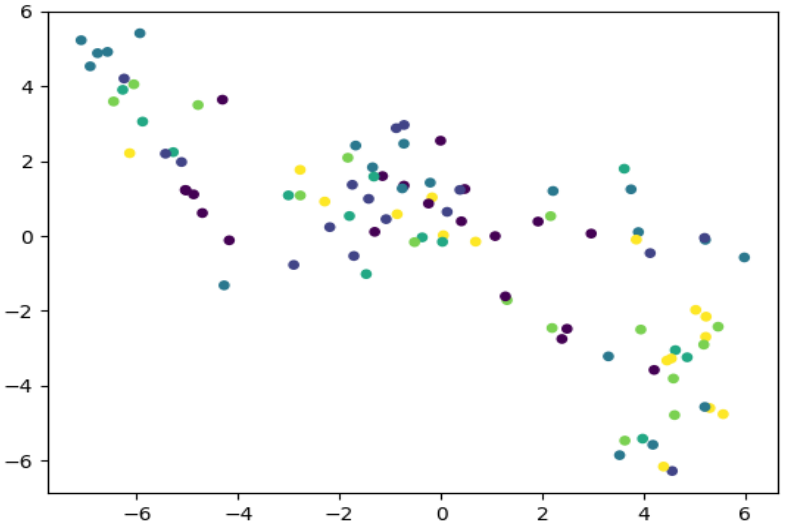
https://www.youtube.com/watch?v=-4MIN-imvck

- t-SNE

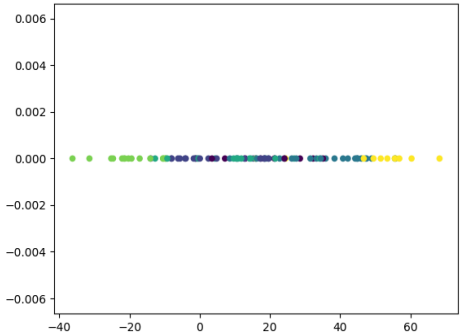
階段一



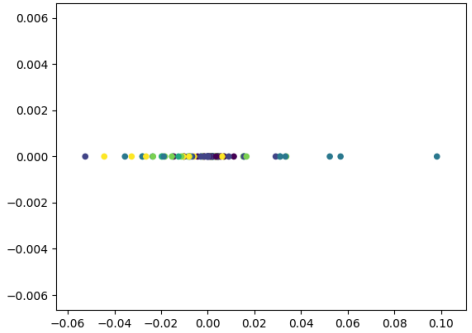
階段二十三



- C的一維散佈圖



階段(II)的C 散佈圖



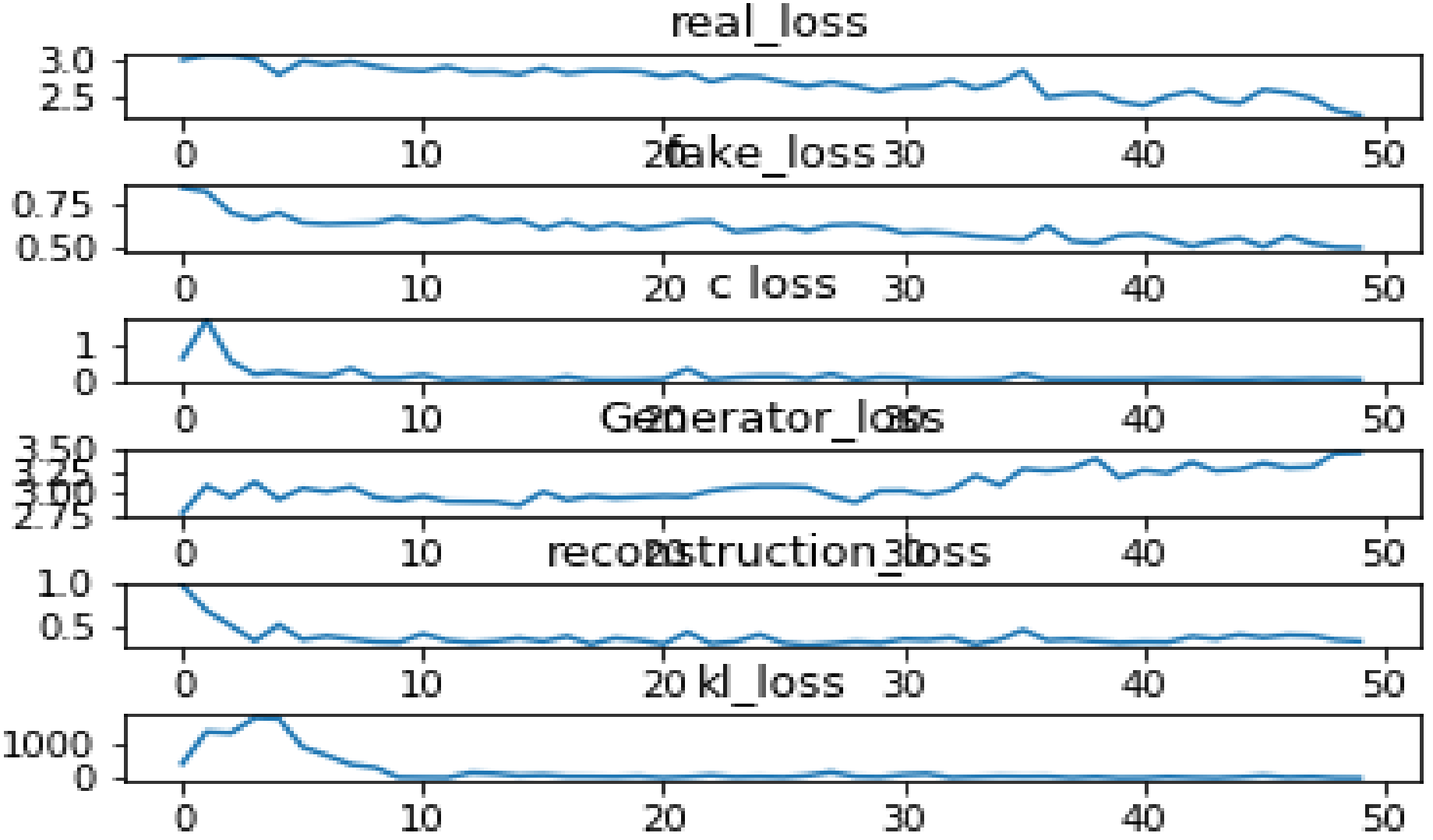
階段(II)+(III)的C散佈圖

- Generator reconstruct 的圖片

G(E(X))

Random sample from Z、C所產生的圖片

- Loss Table


real_loss
3.0
2.5
0
10
20
fake_loss
30
40
50
0.75
0.50
0
10
20
c loss
30
40
50
1
0
0
10
20
Generator_loss
30
40
50
3.50
2.75
0
10
20
30
40
50
1.0
0.5
0
10
20
kl_loss
30
40
50
1000
0
0
10
20
30
40
50

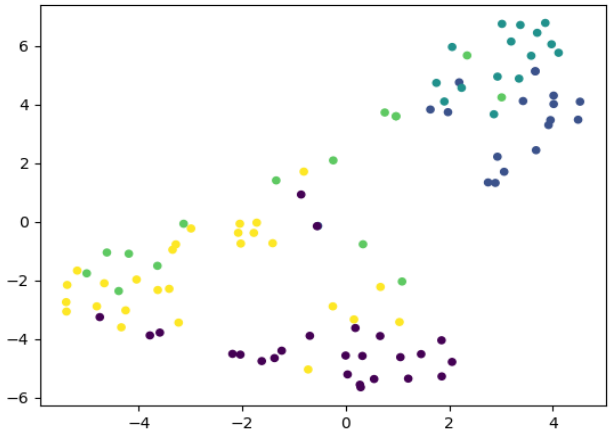
研究結果

- (I) One Direction - What Makes You Beautiful
- 網址如下:

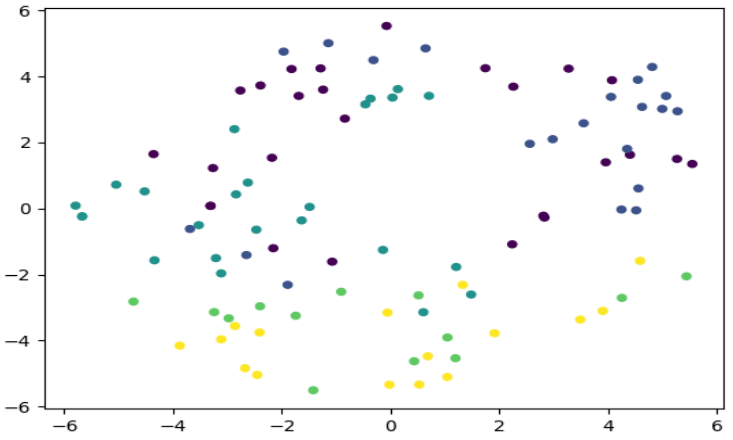
https://www.youtube.com/watch?v=QJO3ROT-A4E

- t-SNE

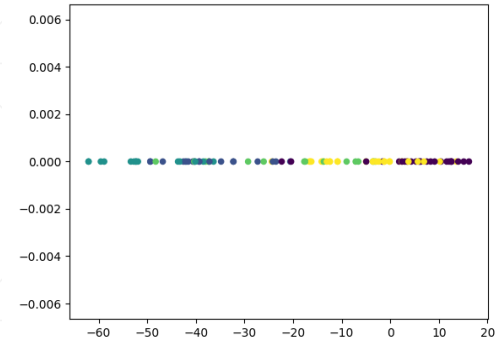
階段一



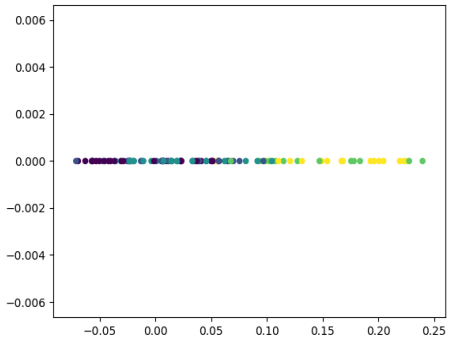
階段二十三



- C的一維散佈圖

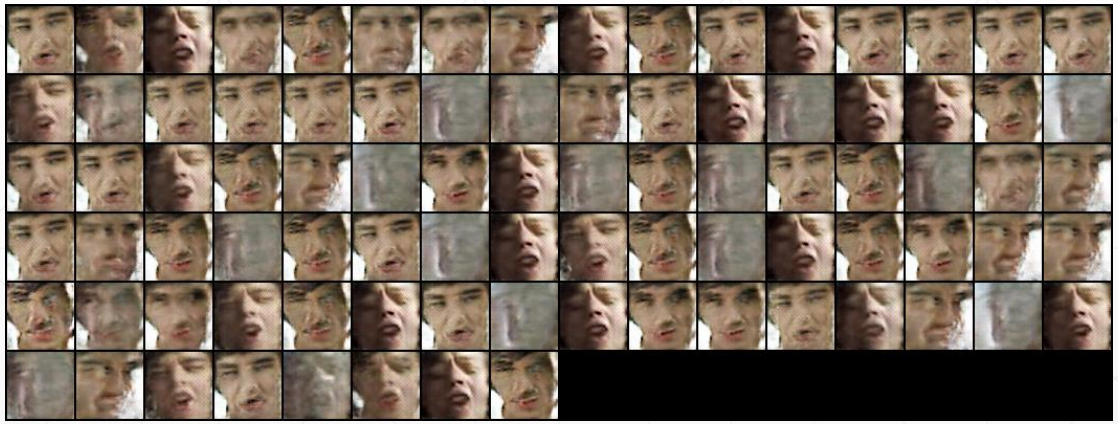


階段(II)的C 散佈圖

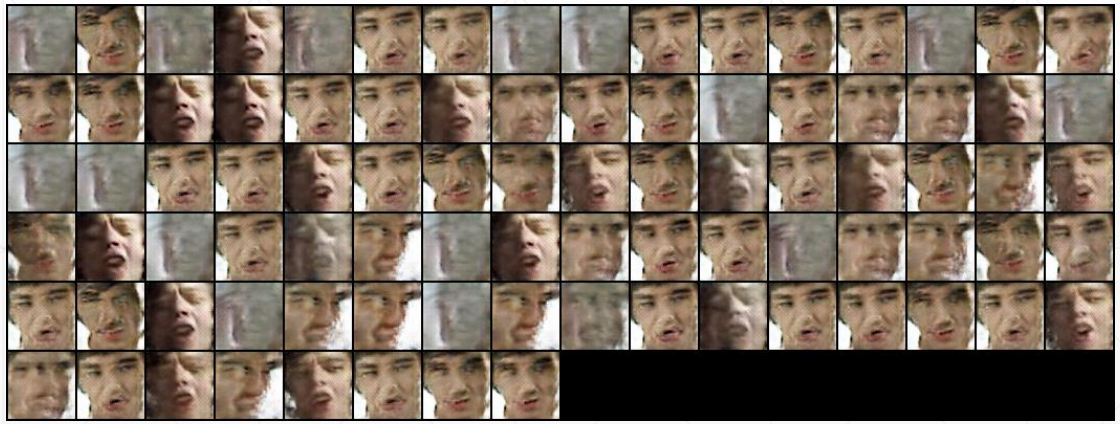


階段(II)+(III)的C散佈圖

- Generator reconstruct 的圖片

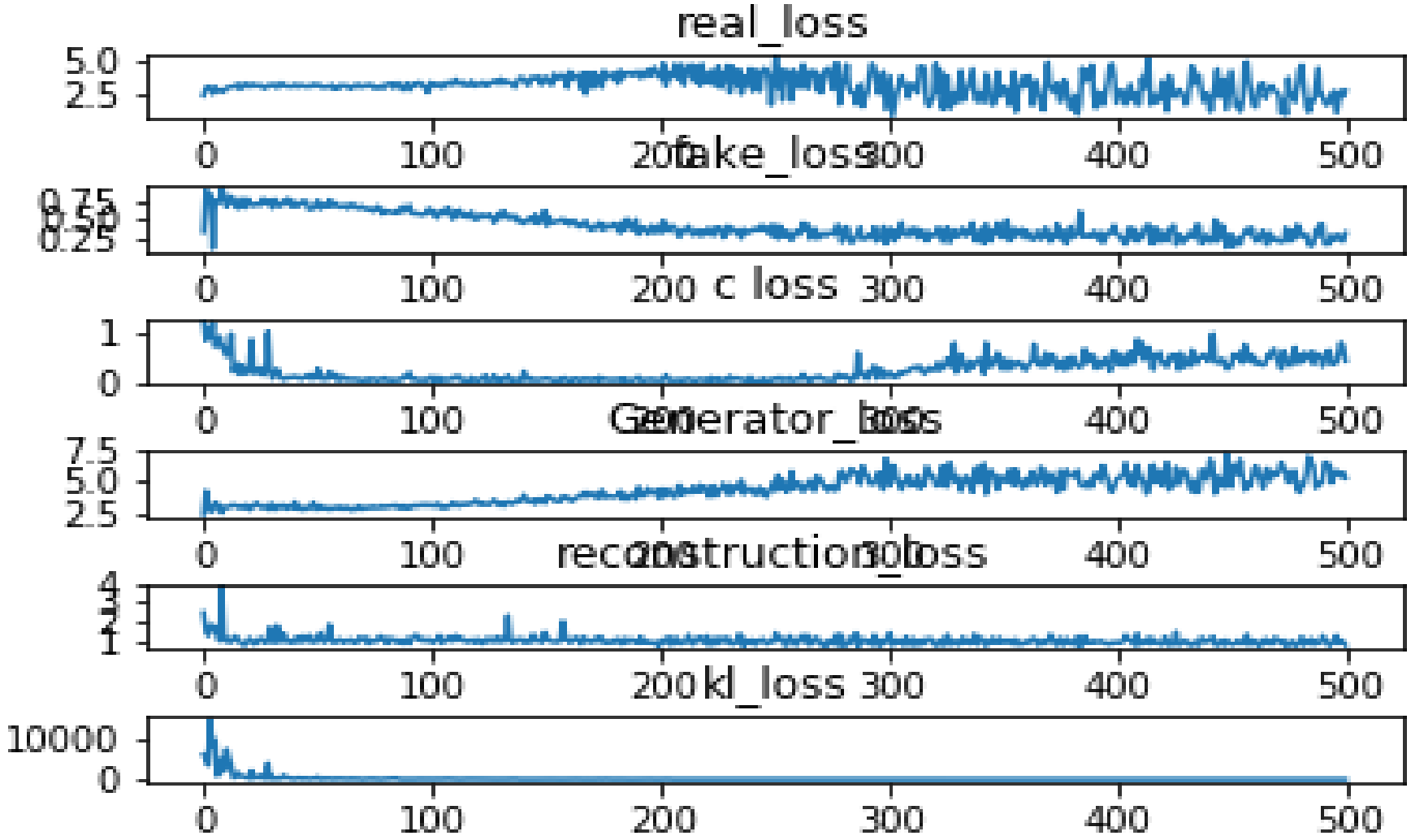


G(E(X))



Random sample from Z、C所產生的圖片

- Loss Table


real_loss
5.0
2.5
0
100
200
300
400
500
fake_loss
0.75
0.50
0.25
0
100
200
c_loss
300
400
500
1
0
0
100
Generator_loss
400
500
7.5
5.0
2.5
0
100
200
reconstruction_loss
300
400
500
0
100
200
kl_loss
300
400
500
10000
0
0
100
200
300
400
500

問題與討論

1.最後的t-SNE看不出分群的效果

A.這方面我們還在思考怎麼做，最直接的就是再多update幾次，但這樣會讓程式執行時間大幅上升，所以我們猜測應該是parameter之間的balance沒有做好，所以之後可以將整個model拆成Encoder+Classifier和InfoGAN兩個部分去分別找各自最好的performance，最終合起來才能找到比較好的結果。

2.Generate的圖片

A. Generater產生的圖片，雖然已經有人臉以及五官的樣子，但目前看來仍不算太像，我們有發現到:產生的圖片五官線條非常明顯，就像畫了很厚的眼線一般，這可能是我們的C所學到的，同樣也需要多tune一陣子來看看結果。

3.Discriminator強過Generator、Classifier強過Encoder

A.從最後的loss table中，我們可以發現這樣的事實，舉例:Discriminator相關的loss不斷下降，代表Discriminator已經進步學得不錯，但Generator Loss卻沒有下降的感覺，我們想到的解法，除了一樣多update幾次，另外一個就是給予noisy label:故意給一些錯的資訊降低學習速度，例如給Discriminator fake image卻說是real image，以此來控制其學習速度。

Thanks for your attention!!