智慧化企業整合
Intelligent Integration of Enterprise

Introduction of Generative Adversarial Network (GAN)

助教:蔡丞洲

Outline

- GAN Introduction
- Condition Generative Adversarial Network
- Intelligent Photo Editing
- Tips to Implement GAN : Feature Extraction
- Tips to Implement GAN : Loss Function
- Demo
- Class Assignment & Homework

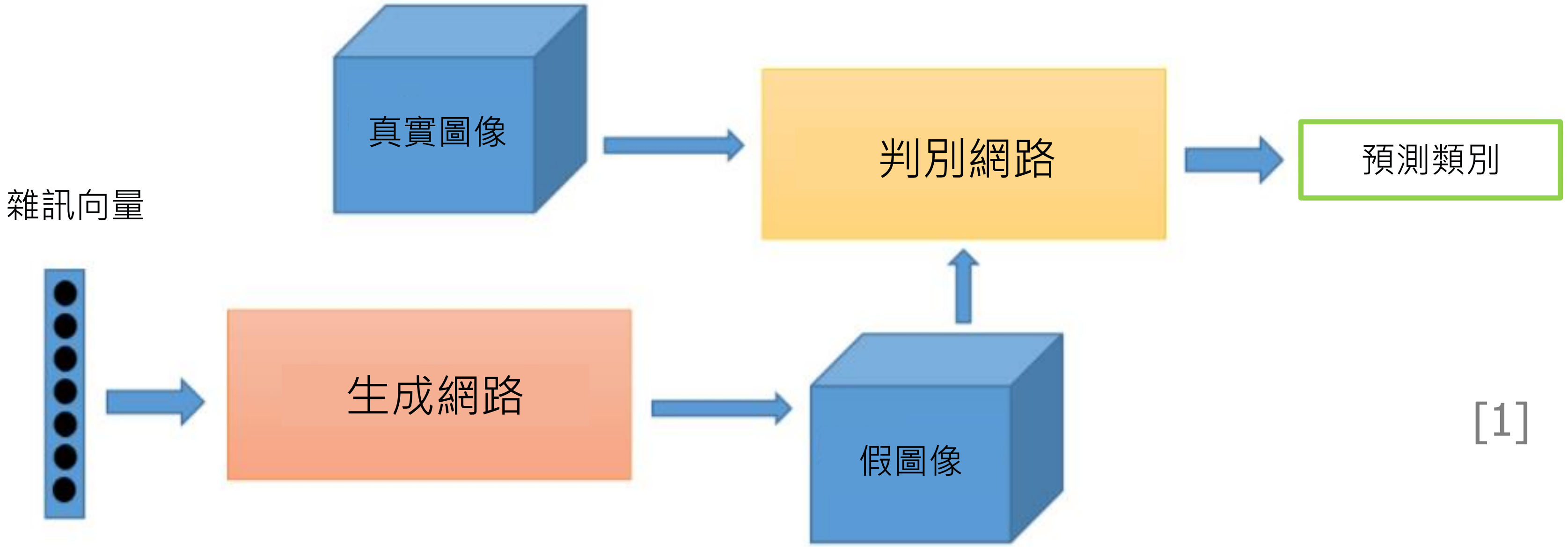
GAN(1/3)

- 生成對抗網路 (GAN) 是 2014 年 蒙 特 婁 大 學 博 士 生 Ian Goodfellow 提出來的。
- 相較傳統的模型，他存在兩個不同的網路，而不是單一的網路，並且訓練方式採用的是對抗訓練方式。
- GAN經由**小量**真實資料，產生**大量**的訓練資料，作為非監督學習的重要訓練方法。

[1]

GAN(2/3)

- 主要概念為一個是偽造者，他不斷製造假鈔，另一個是警察，不斷從偽造者那邊拿到假鈔，判斷是真或假，偽造者就根據警察判斷結果的回饋，不斷改良，最後假鈔變成真假難辨。
- 在GAN架構下，偽造者就稱為生成模型（generative model），警察稱為判別模型（discriminative model)。



[1]

GAN(3/3)

- 處理資料類型眾多，如**圖像**與**影音**的生成、合成、辨識、修復等等，進階一點的則是輸入文本描述便能生成與形容相符的圖像，或者透過語言模型實現機器翻譯等。

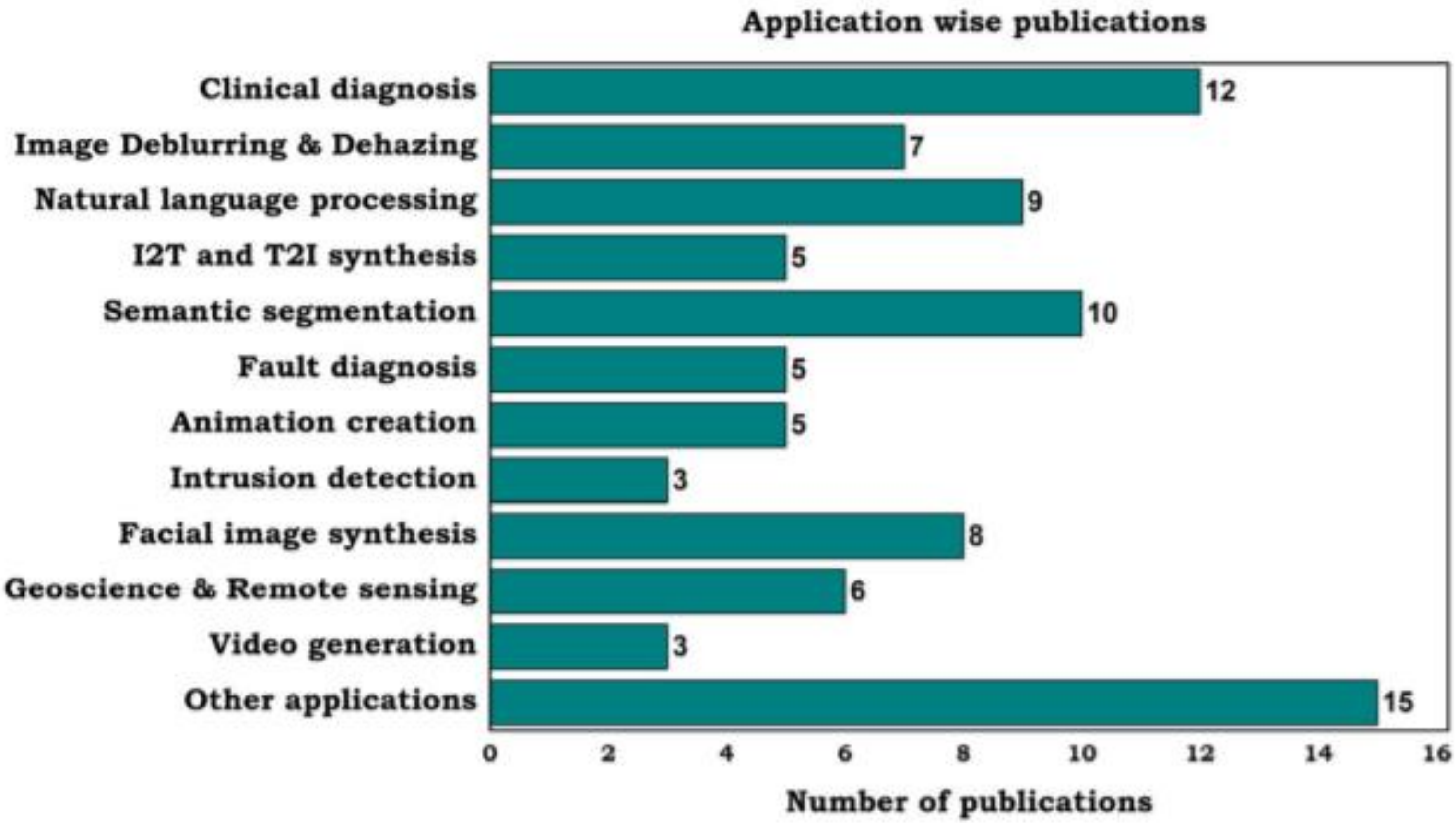
應用(1/2)

圖像超解析

圖片去霧化

[2]

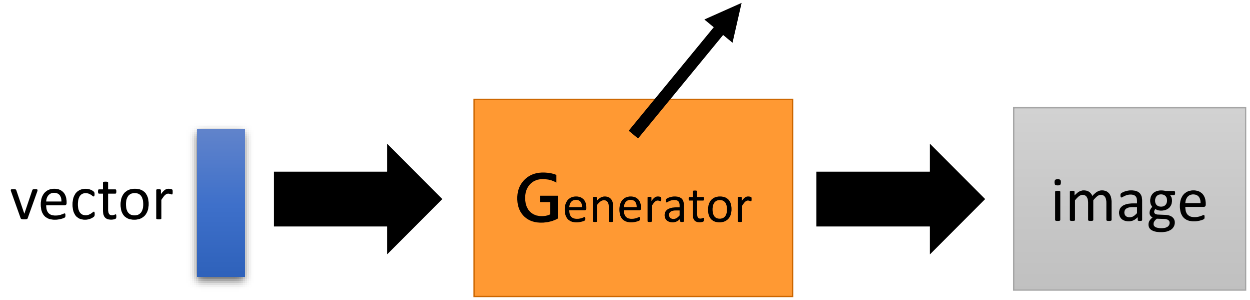
應用(2/2)


Application wise publications
Clinical diagnosis
Image Deblurring & Dehazing
Natural language processing
I2T and T2I synthesis
Semantic segmentation
Fault diagnosis
Animation creation
Intrusion detection
Facial image synthesis
Geoscience & Remote sensing
Video generation
Other applications
12
7
9
5
10
5
5
3
8
6
3
15
0
2
4
6
8
10
12
14
16
Number of publications

[2]

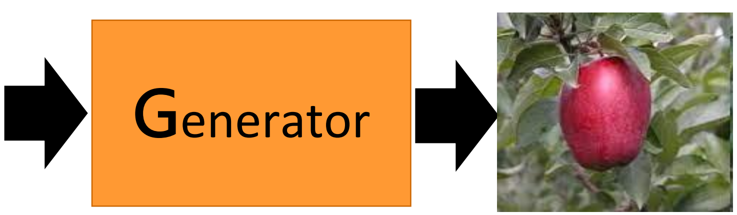
生成器

生成器為神經網路，可做一函數。

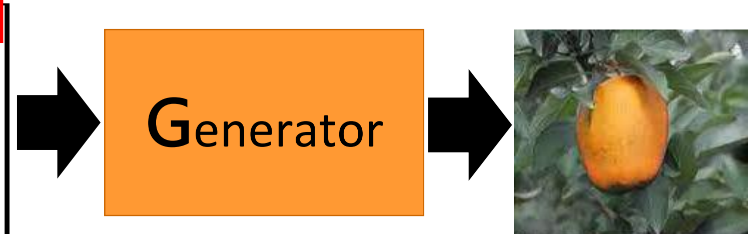


多維矩陣

$$\begin{bmatrix}0.1\\-3\\\vdots\\2.4\\0.9\end{bmatrix}$$



$$\begin{bmatrix}\boxed{3}\\-3\\\vdots\\2.4\\0.9\end{bmatrix}$$



- 輸入向量的每個維度代表一些特徵。

[3]

判別器

判別器為神經網路，可做一函數。

image → Discriminator → scalar

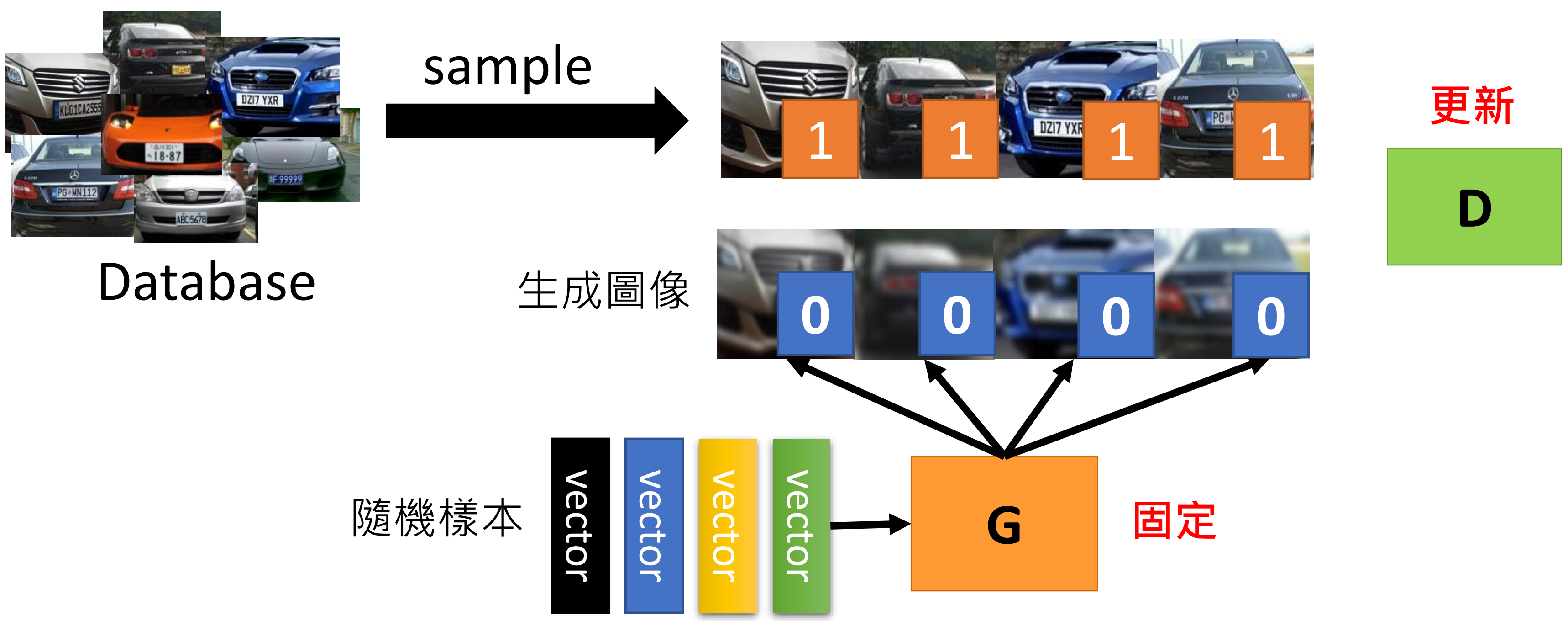
值越大越為真
越值小越為假

 → Discriminator → 1.0

 → Discriminator → 0.1

訓練(1/3)

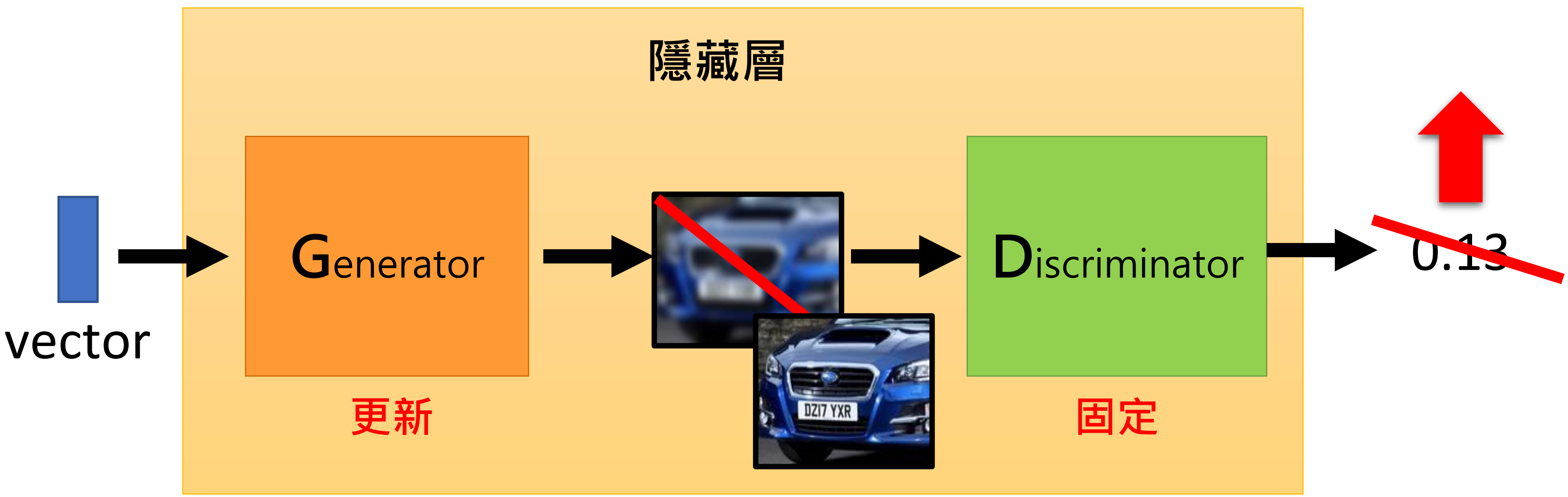
Step 1 : 固定生成器 G，更新判別器 D



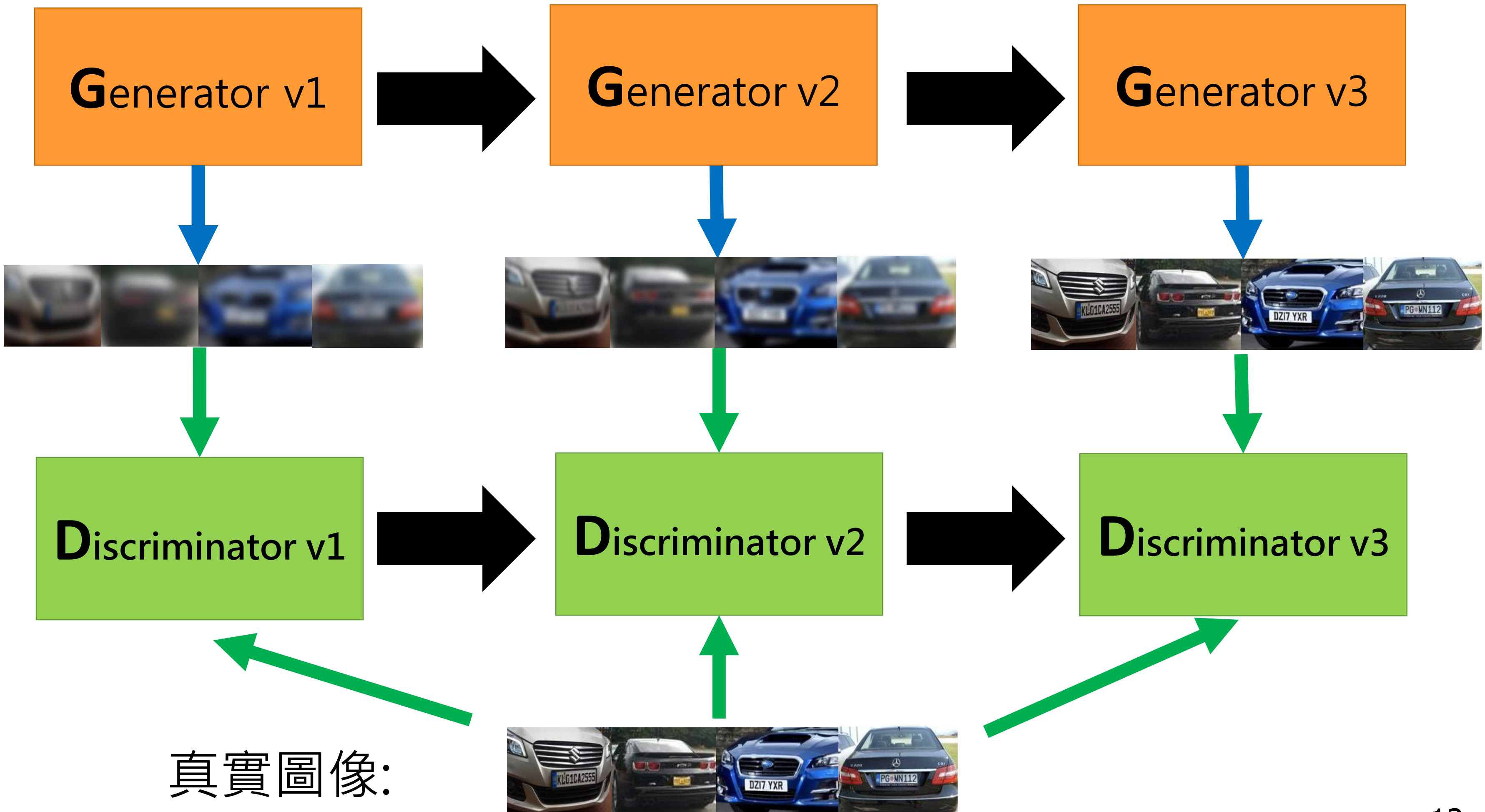
- 鑑別器學習將高分分配給真實對象，將低分分配給生成的對象。

訓練(2/3)

Step 2：固定判別器 D，更新生成器G



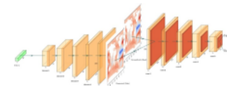
訓練(3/3)


Generator v1
Generator v2
Generator v3
Discriminator v1
Discriminator v2
Discriminator v3

真實圖像:

演進

2014

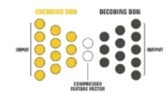
GAN [1] 

GAN framework proposed

 **CGAN** [4]

Label specific sample generation

2015

VAE-GAN [14] 

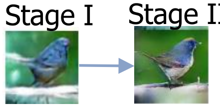
Traditional GAN feature extraction solution

2016

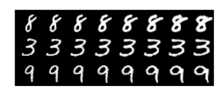
LSGAN [11] 

Alternative to traditional GAN training

Stage I Stage II

 **StackGAN** [5]

Text to image synthesis

InfoGAN [13] 

Traditional GAN feature extraction solution

 **SRGAN** [15]

Image super resolution

2017

 **WGAN** [12]

Alternative to traditional GAN training

Cycle GAN &Disco GAN & Dual GAN [5.8.9]

Unpaired image to image translation

 **StarGAN** [10]

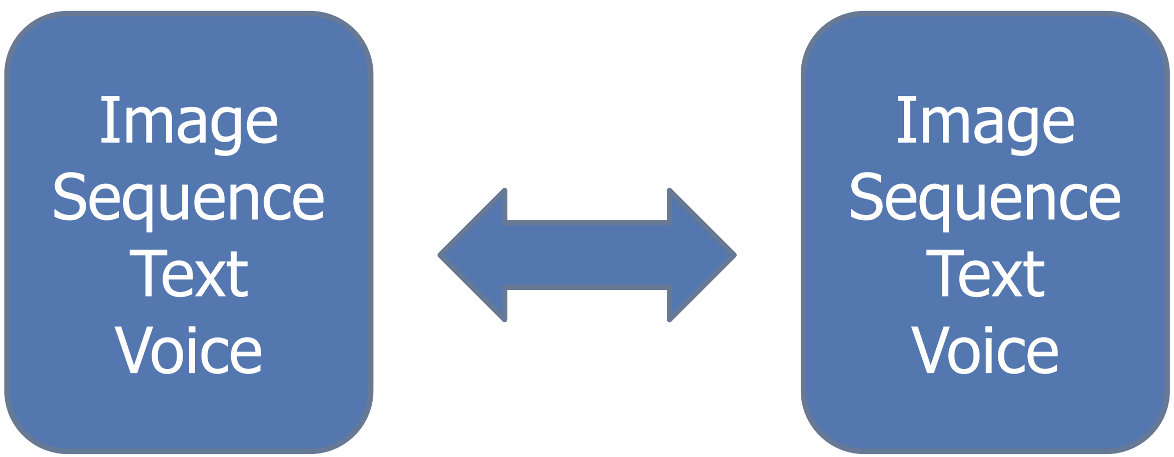
Multi-domain image to image translation

2018

Condition Generative Adversarial Network

CGAN

- CGAN(Conditional Generative Adversarial Nets)
- 透過加入一些條件信息來控制GAN生成的圖片，而不是單純的隨機生成圖片



[4]

Text-to-Image (1/2)

- **傳統監督學習方法**

c : a dog is running → **NN** → Image ↔ 越像越好

輸入: “car”

一個模糊的圖像！

Text-to-Image (2/2)

c: car → G

Normal distribution z → G

G → Image x = G(c,z)

c → D (better)

x → D (better)

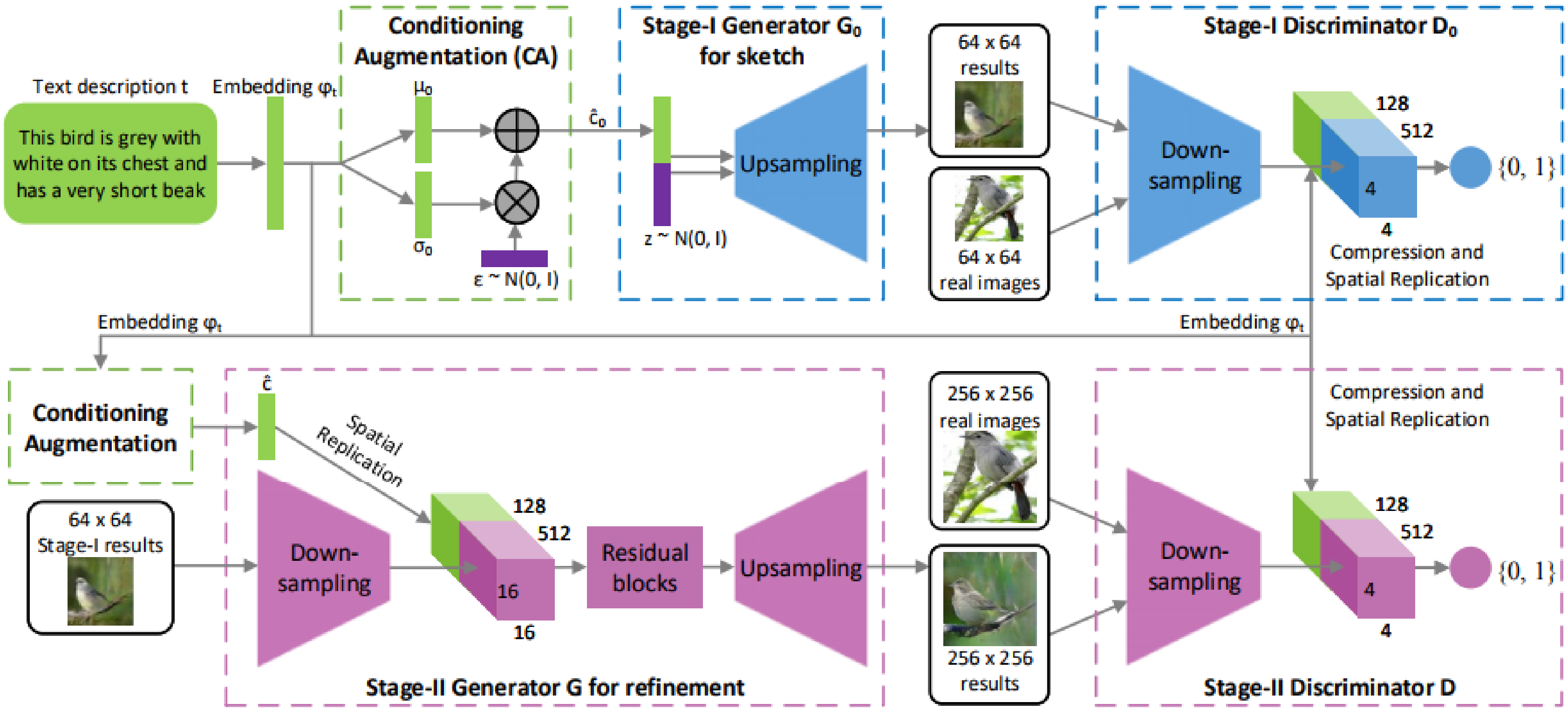
D (better) → scalar → 判別x是否為真實圖像

+ c和x是否成對

正確成對: (car , ) 1

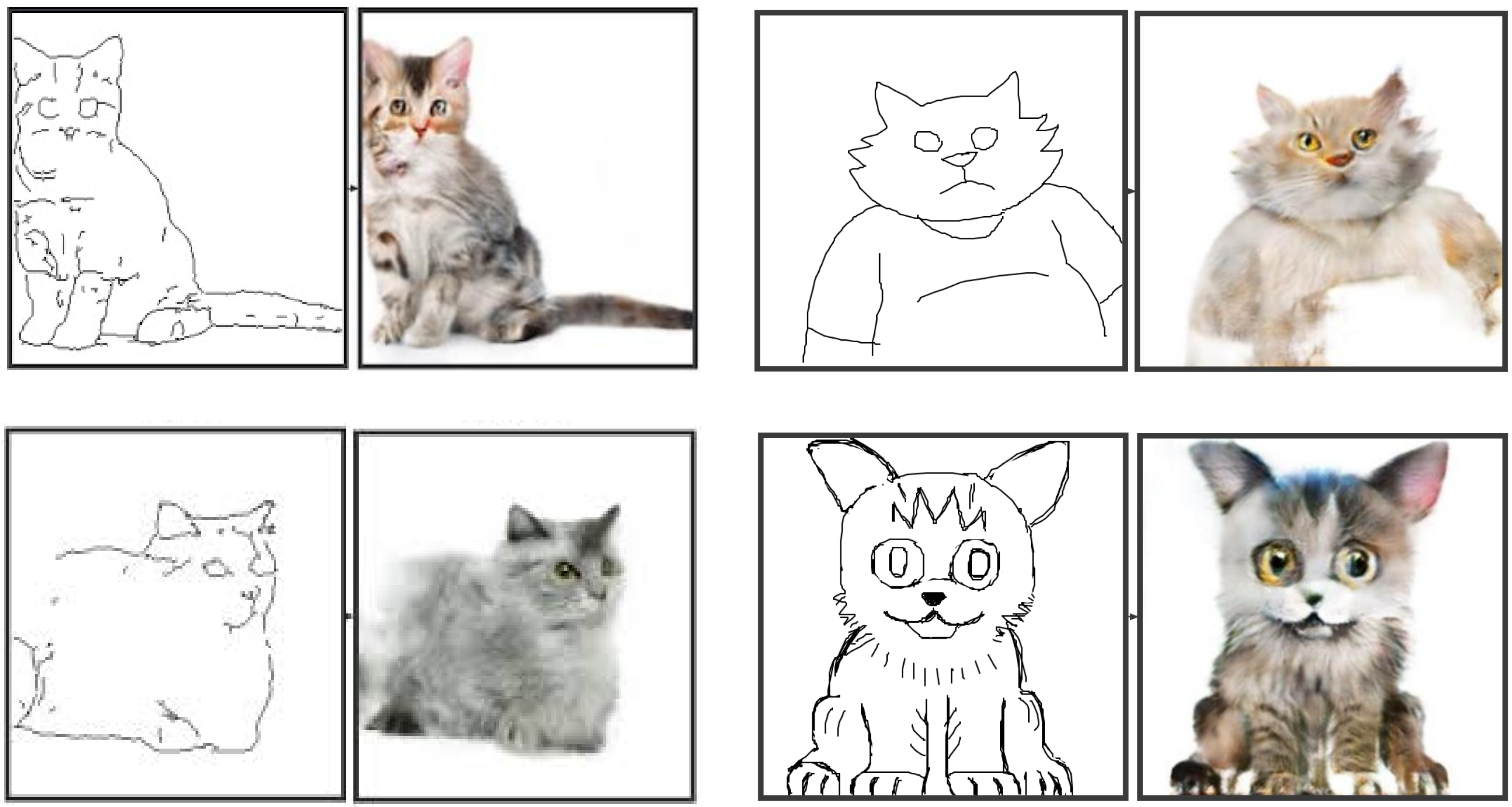
錯誤成對: (cat , ) 0 (car , Image) 0

Stack GAN


Text description t
This bird is grey with white on its chest and has a very short beak
Embedding φt
Conditioning Augmentation (CA)
μ0
σ0
ε ~ N(0, I)
ĉ0
Stage-I Generator G0 for sketch
z ~ N(0, I)
Upsampling
64 x 64 results
64 x 64 real images
Stage-I Discriminator D0
Down-sampling
128
512
4
4
Compression and Spatial Replication
{0, 1}
Embedding φt
Embedding φt
Conditioning Augmentation
ĉ
Spatial Replication
64 x 64 Stage-I results
Down-sampling
128
512
16
16
Residual blocks
Upsampling
Stage-II Generator G for refinement
256 x 256 real images
256 x 256 results
Compression and Spatial Replication
Down-sampling
128
512
4
4
{0, 1}
Stage-II Discriminator D

[5]

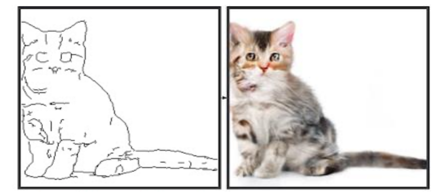
Image-to-image (1/2)

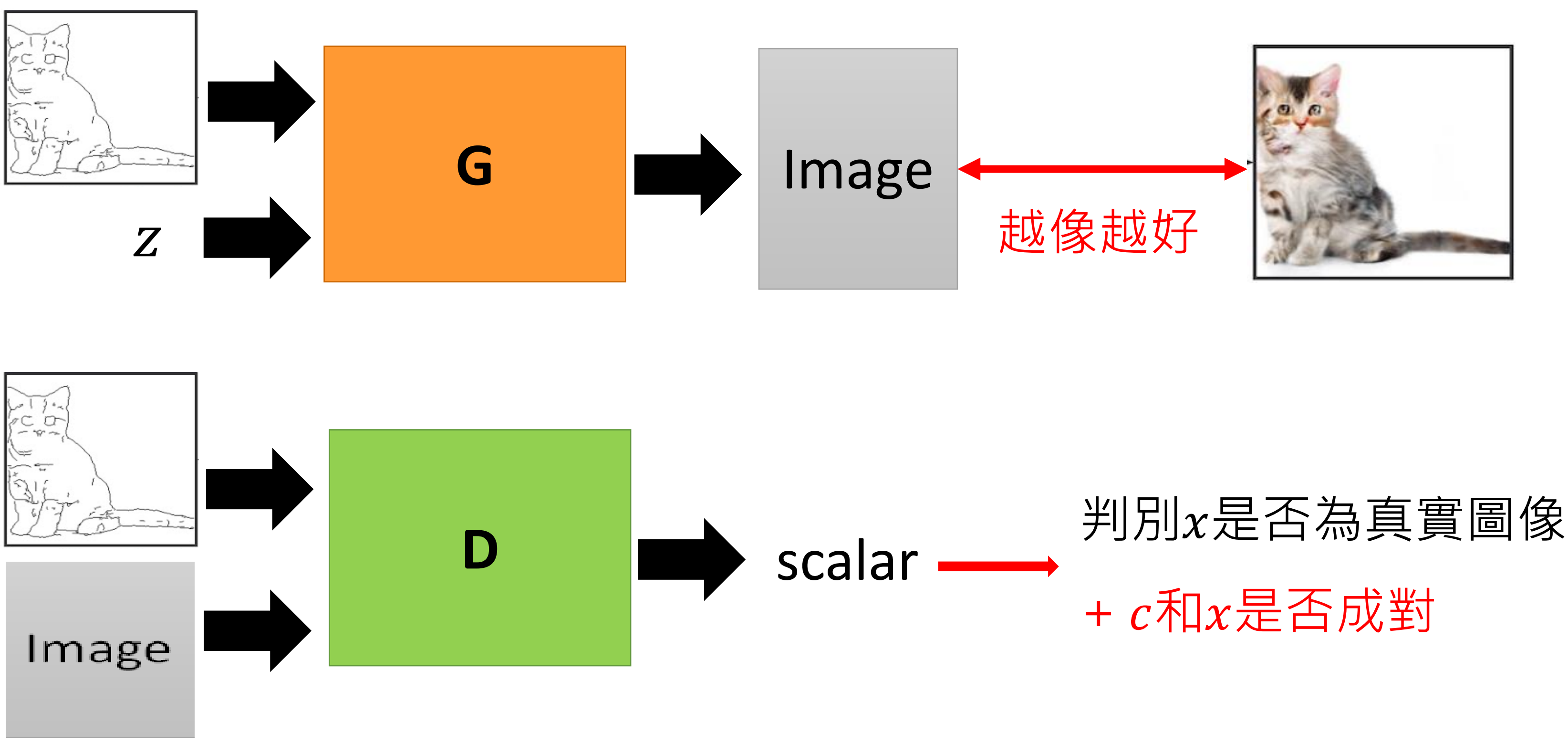


[6]

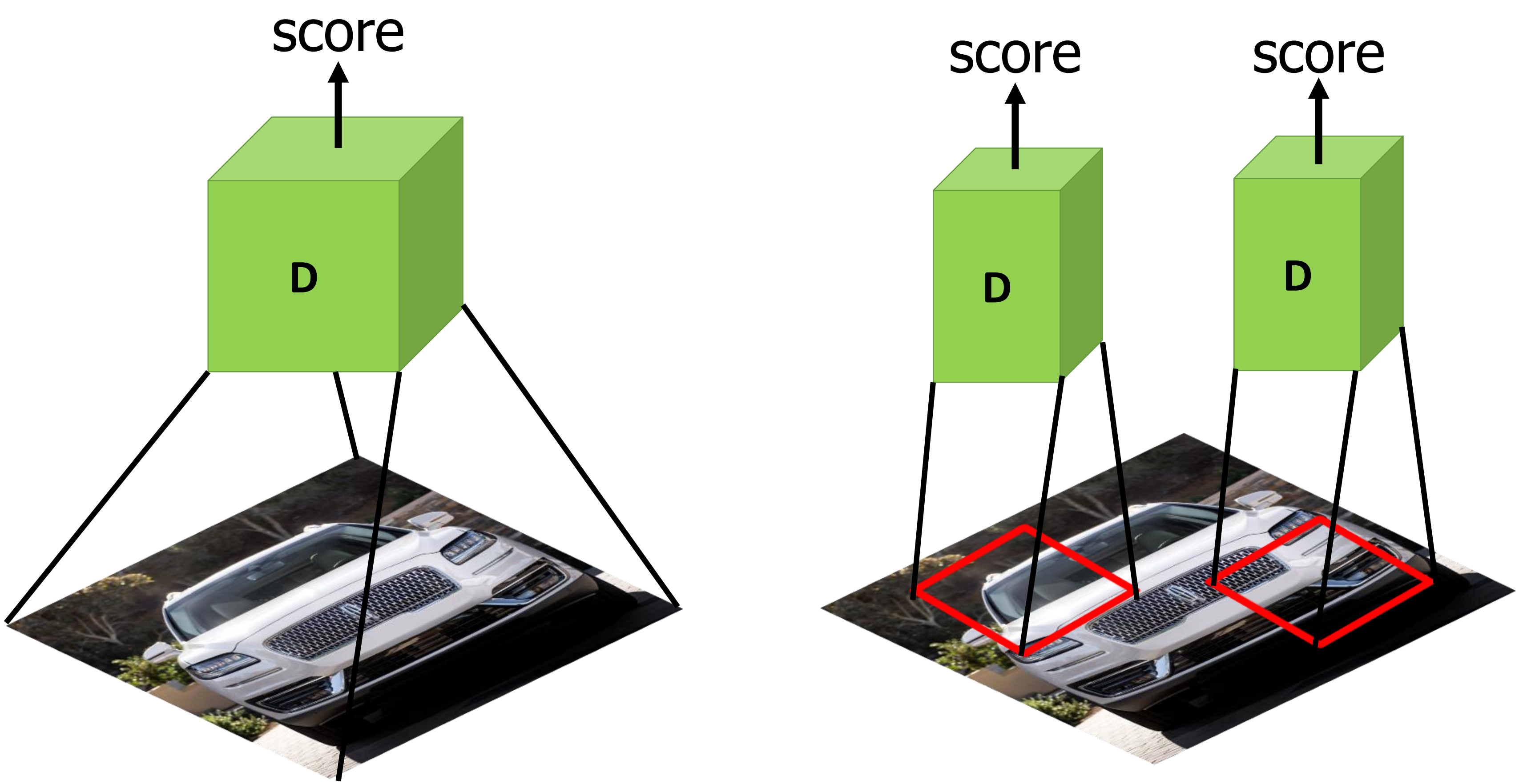
Image-to-image (2/2)

成對資料:



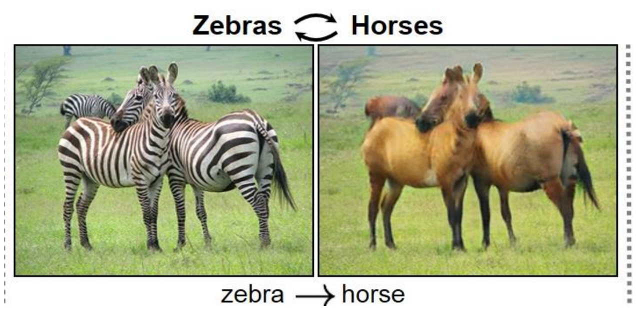


Patch GAN



Cycle GAN(1/4)

- Cycle GAN
- 由朱俊彥於2017年提出的非監督生成對抗網路
- 監督式學習須以成對圖像作為輸入，但在現實中，很多任務是無法取得成對訓練數據
- 不需要成對資料，只需要蒐集兩種不同類型的數據集，去學習數據域A和B的之間的關係，
- 輸入數據域A轉換為數據域B

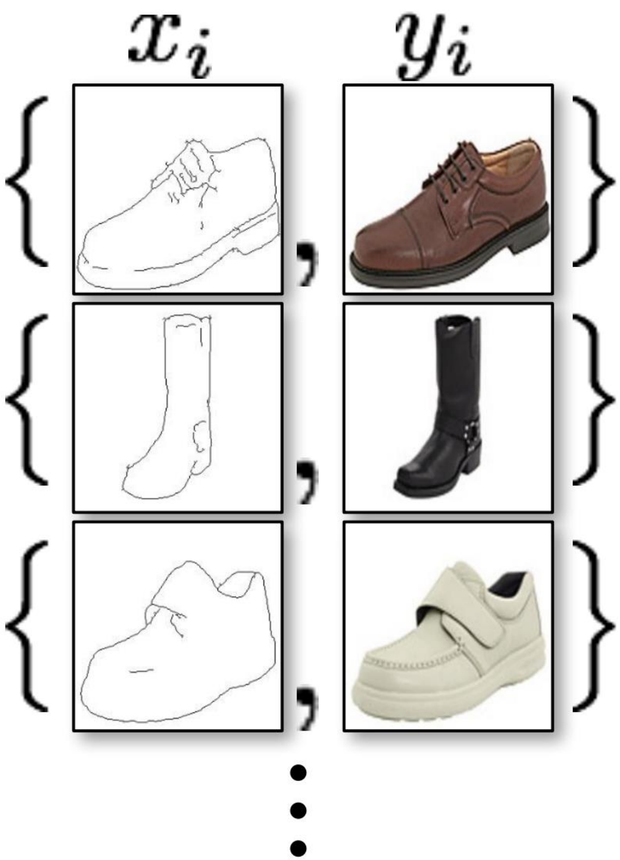


[7]

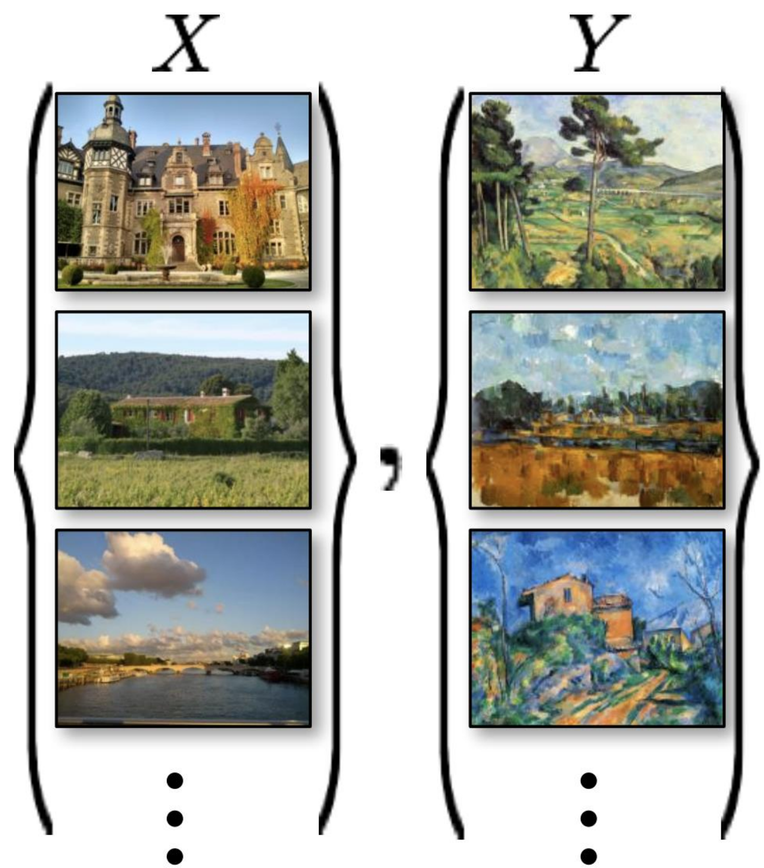
Cycle GAN (朱俊彦,2017) (2/4)

Cycle GAN (3/4)

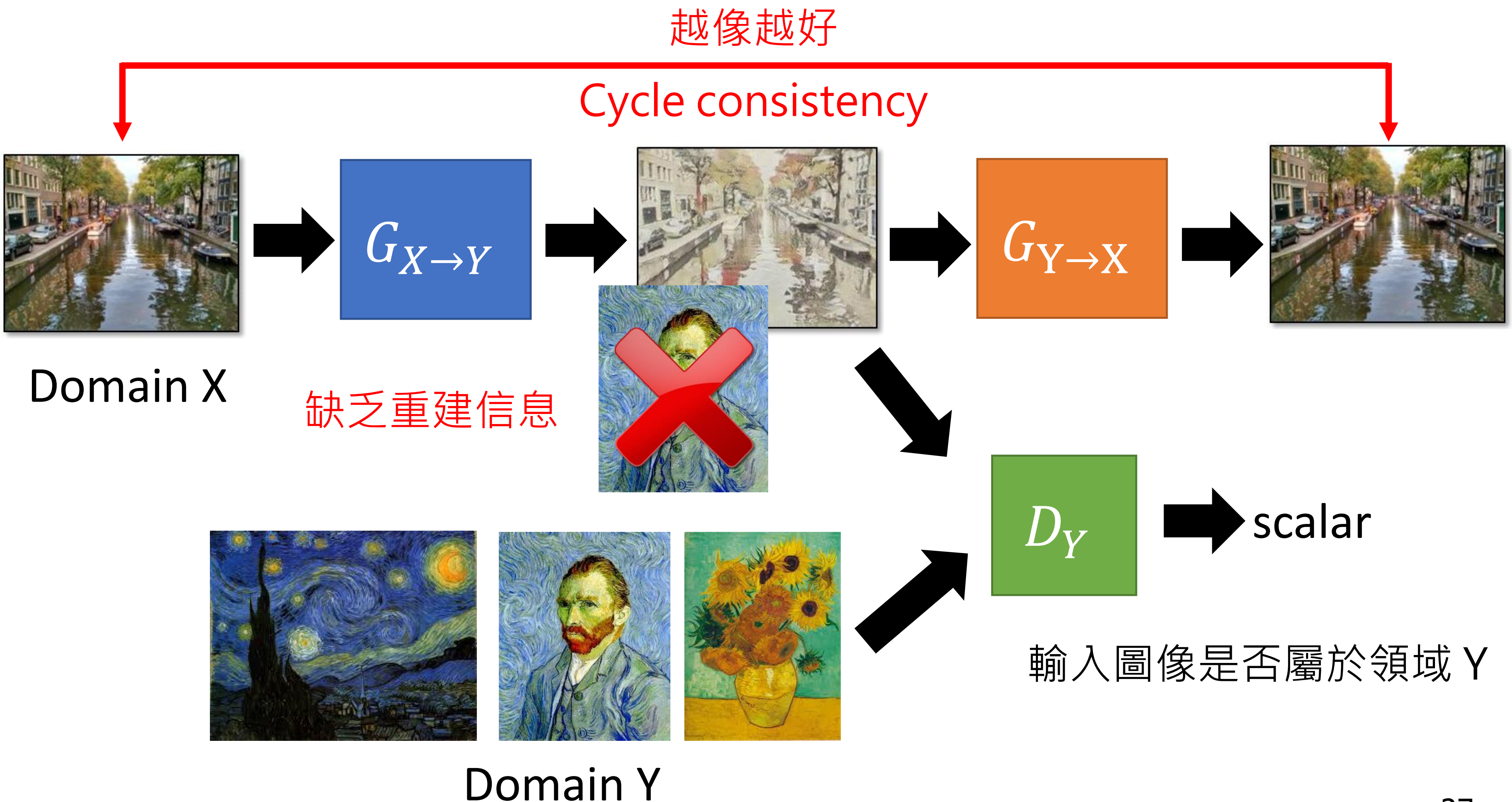
成對



非成對

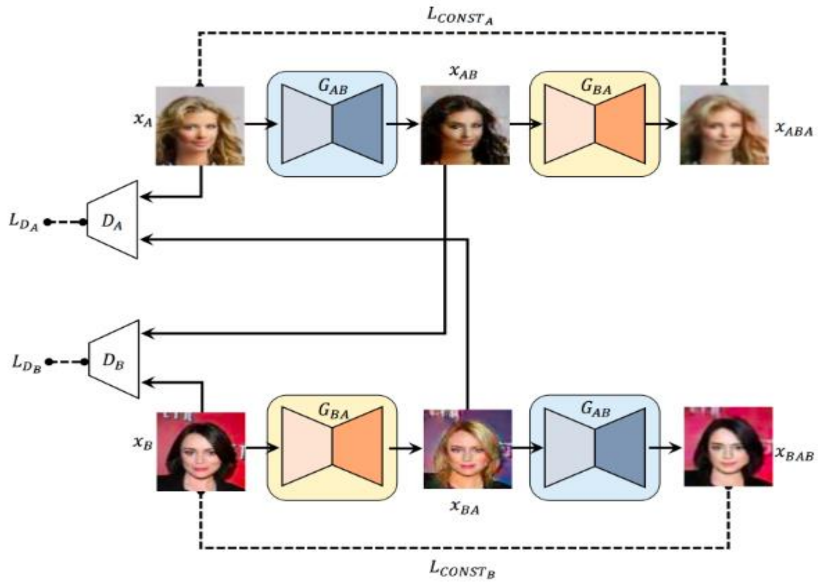


Cycle GAN (4/4)


越像越好
Cycle consistency
$G_{X \rightarrow Y}$
$G_{Y \rightarrow X}$
Domain X
缺乏重建信息
D_Y
scalar
輸入圖像是否屬於領域 Y
Domain Y

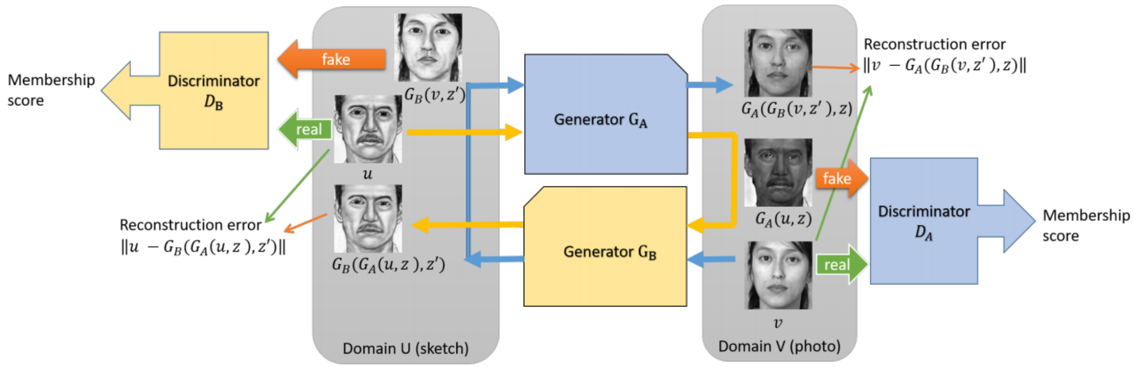
Disco GAN & Dual GAN

Disco GAN



[8]

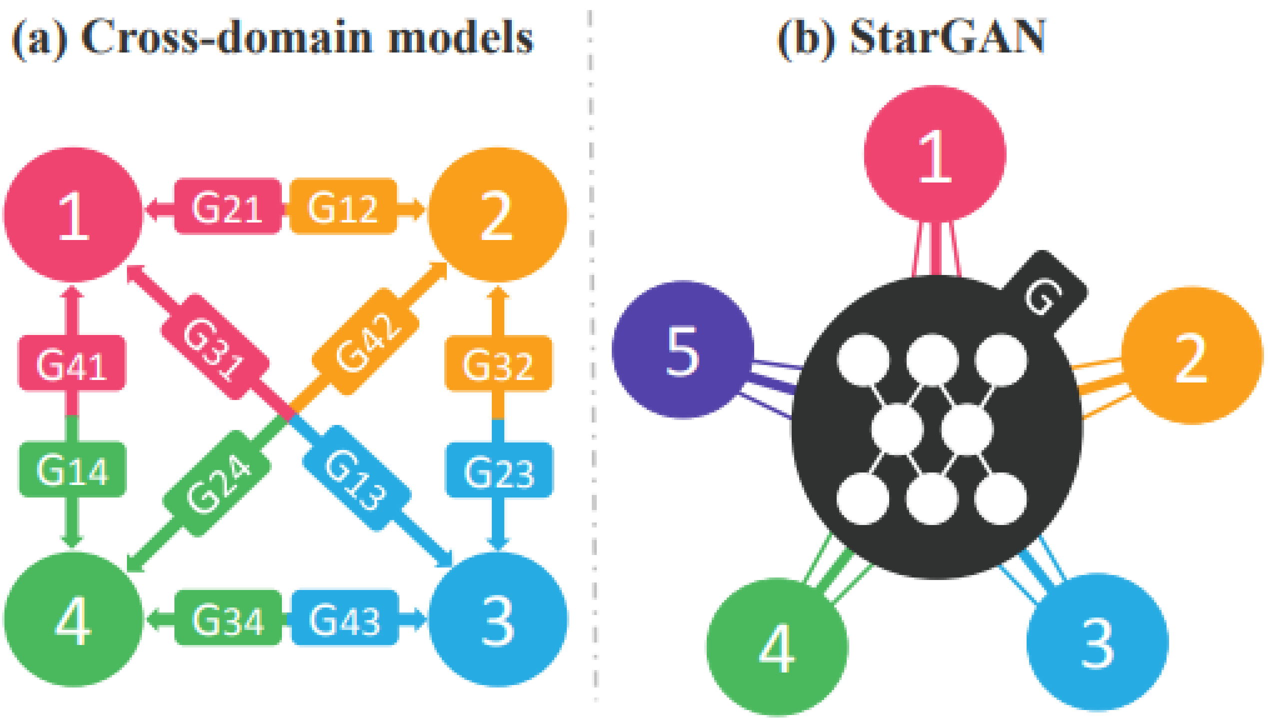
Dual GAN



[9]

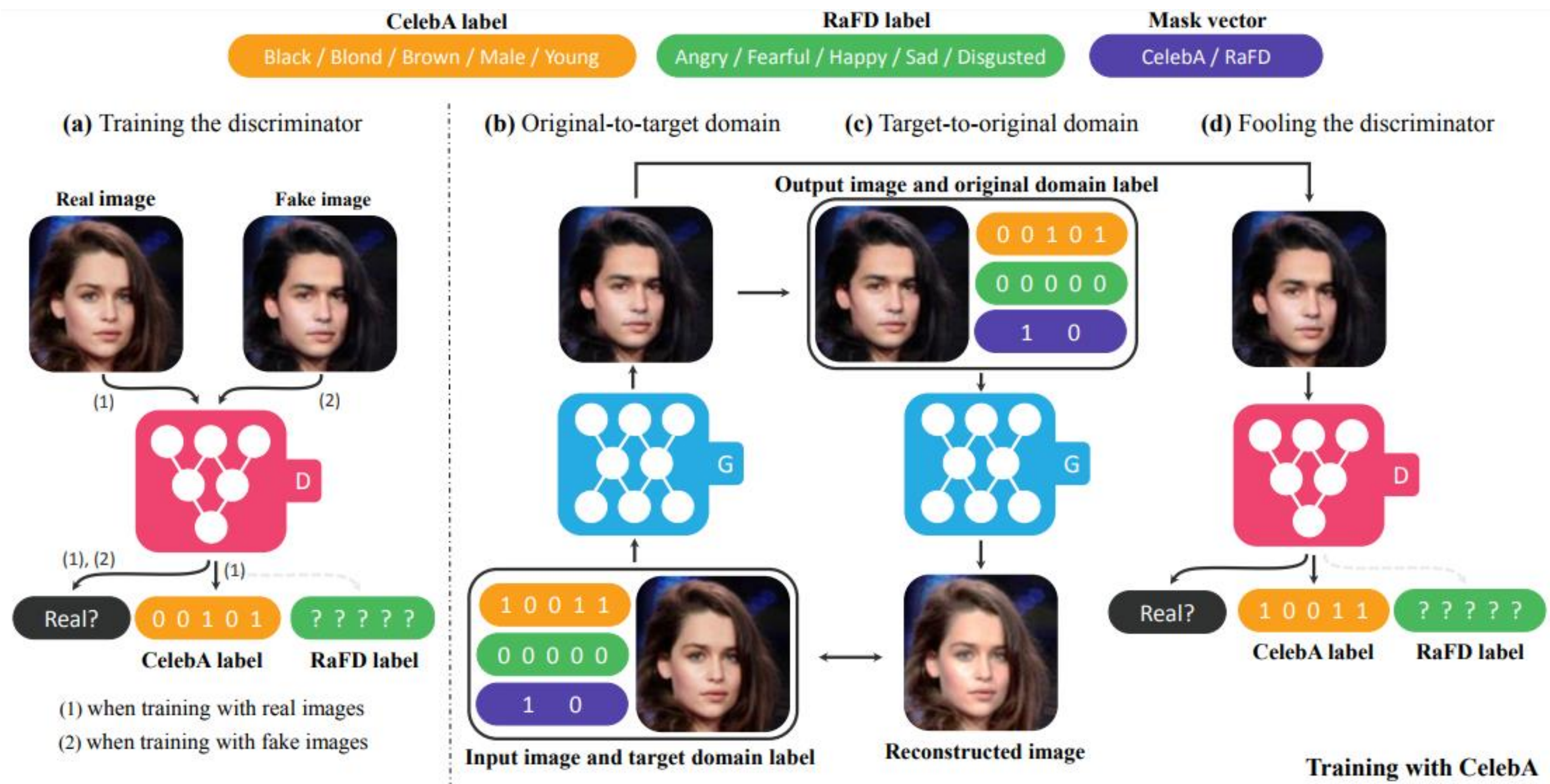
StarGAN(1/2)

- 應用於多個領域



[10]

StarGAN (2/2)



CelebA label
Black / Blond / Brown / Male / Young
RaFD label
Angry / Fearful / Happy / Sad / Disgusted
Mask vector
CelebA / RaFD
(a) Training the discriminator
(b) Original-to-target domain
(c) Target-to-original domain
(d) Fooling the discriminator
Real image
Fake image
(1)
(2)
D
(1), (2)
(1)
Real?
0 0 1 0 1
? ? ? ? ?
CelebA label
RaFD label
(1) when training with real images
(2) when training with fake images
Output image and original domain label
0 0 1 0 1
0 0 0 0 0
1 0
G
G
1 0 0 1 1
0 0 0 0 0
1 0
Input image and target domain label
Reconstructed image
D
Real?
1 0 0 1 1
? ? ? ? ?
CelebA label
RaFD label
Training with CelebA

Intelligent Photo Editing

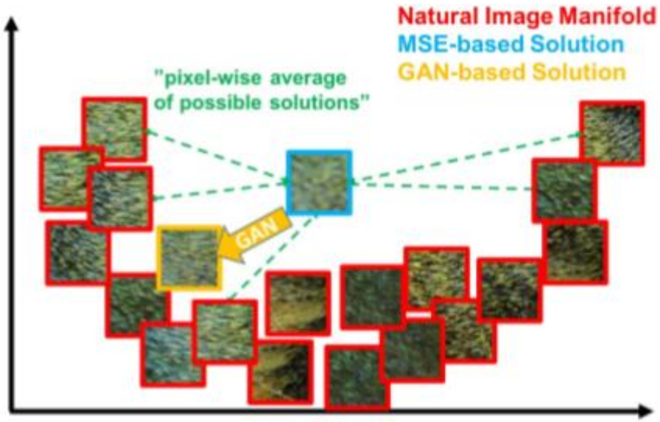
Image super resolution

- 低解析度影象重建出相應的高解析度影象

[15]

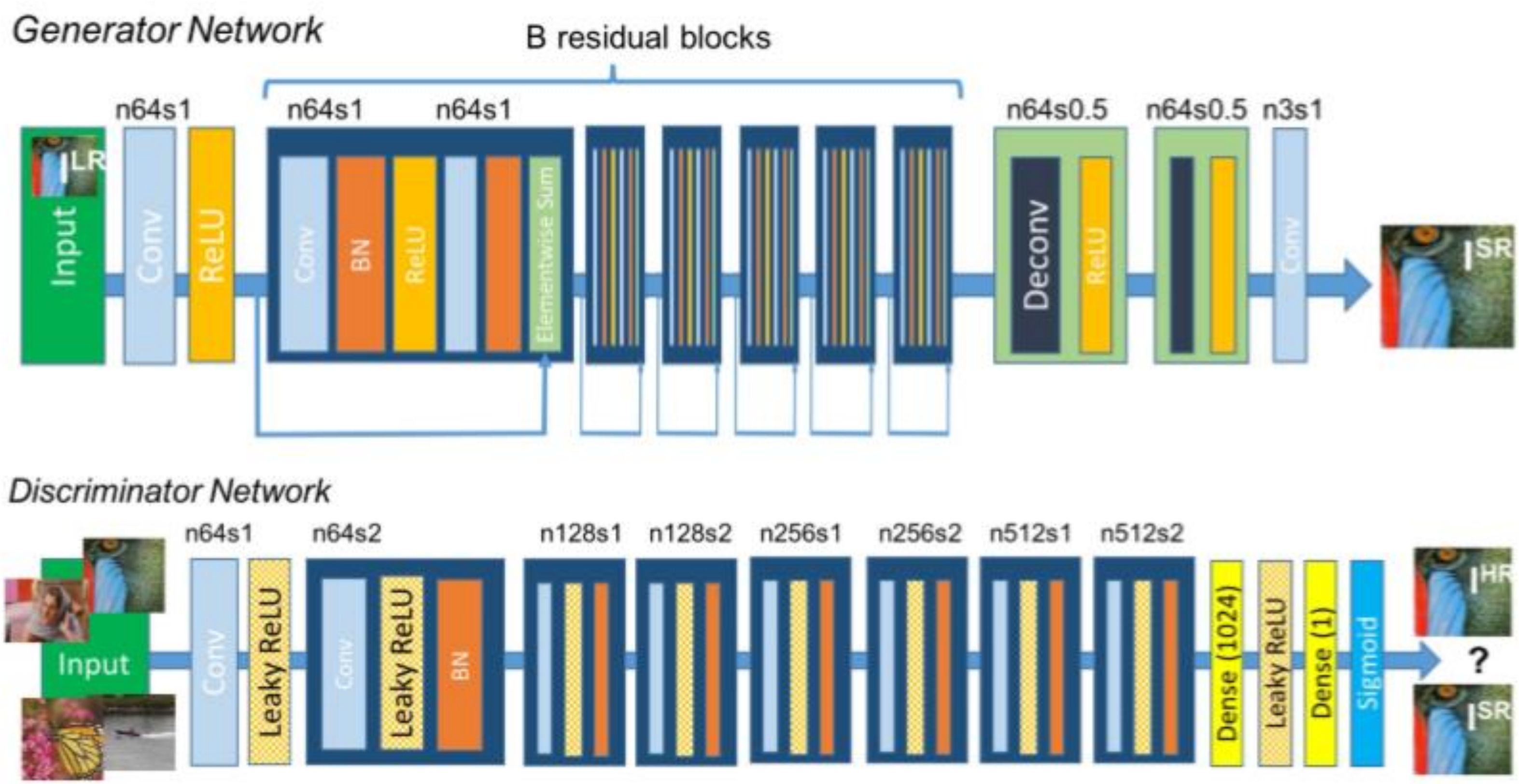
SRGAN (1/2)

- SRGAN(Super-Resolution GAN)
- 傳統的NN重建的超解析圖像中經常缺少紋理細節。
- 多以均方誤差 (MSE)作為損失函數，使得生成圖像較平滑、質量較差。
- 學習模糊圖像及清楚圖像之間的關係，並透過生成器來生成超解析度圖像。



[15]

SRGAN(2/2)

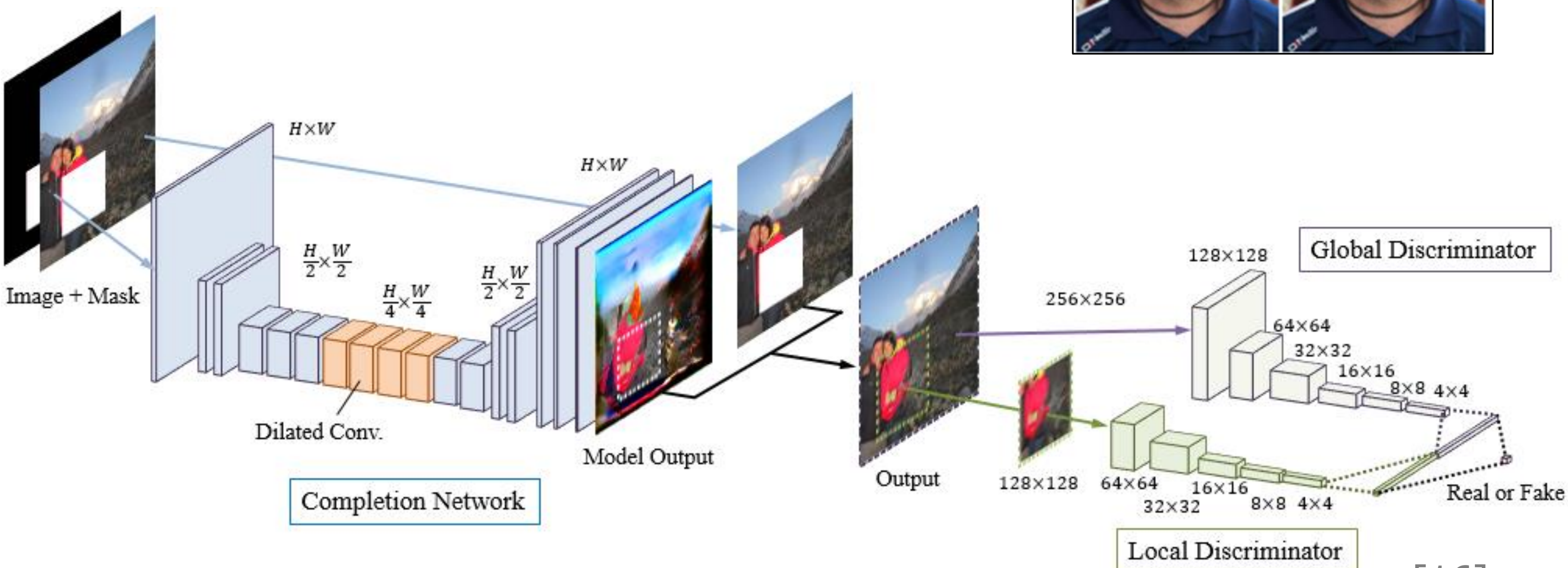

Generator Network
B residual blocks
n64s1
n64s1
n64s1
n64s0.5
n64s0.5
n3s1
Input
Conv
ReLU
Conv
BN
ReLU
Elementwise Sum
Deconv
ReLU
Conv
I^{LR}
I^{SR}
Discriminator Network
n64s1
n64s2
n128s1
n128s2
n256s1
n256s2
n512s1
n512s2
Input
Conv
Leaky ReLU
Conv
Leaky ReLU
BN
Dense (1024)
Leaky ReLU
Dense (1)
Sigmoid
I^{HR}
?
I^{SR}

[15]

Image completion (1/2)

[16]

Image completion(2/2)



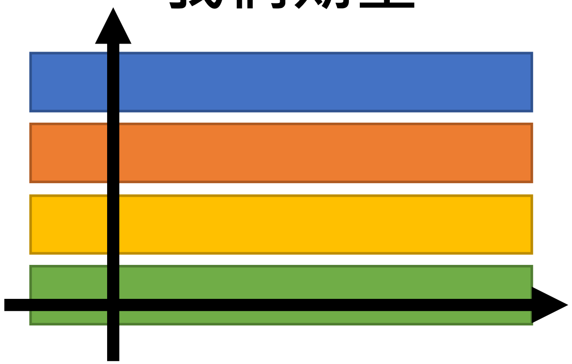
[16]

Tips to Implement GAN : Feature Extraction

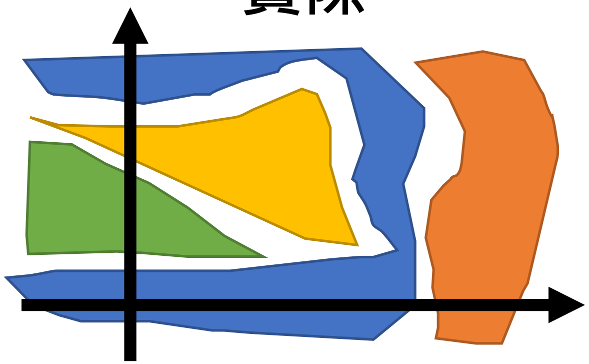
InfoGAN(1/4)

- **問題**：無法通過控制雜訊z的某些維度來控制生成數據的語義特徵
- **主要任務**：特徵提取
- **目的**：解決輸入特徵不明確的問題

我們期望



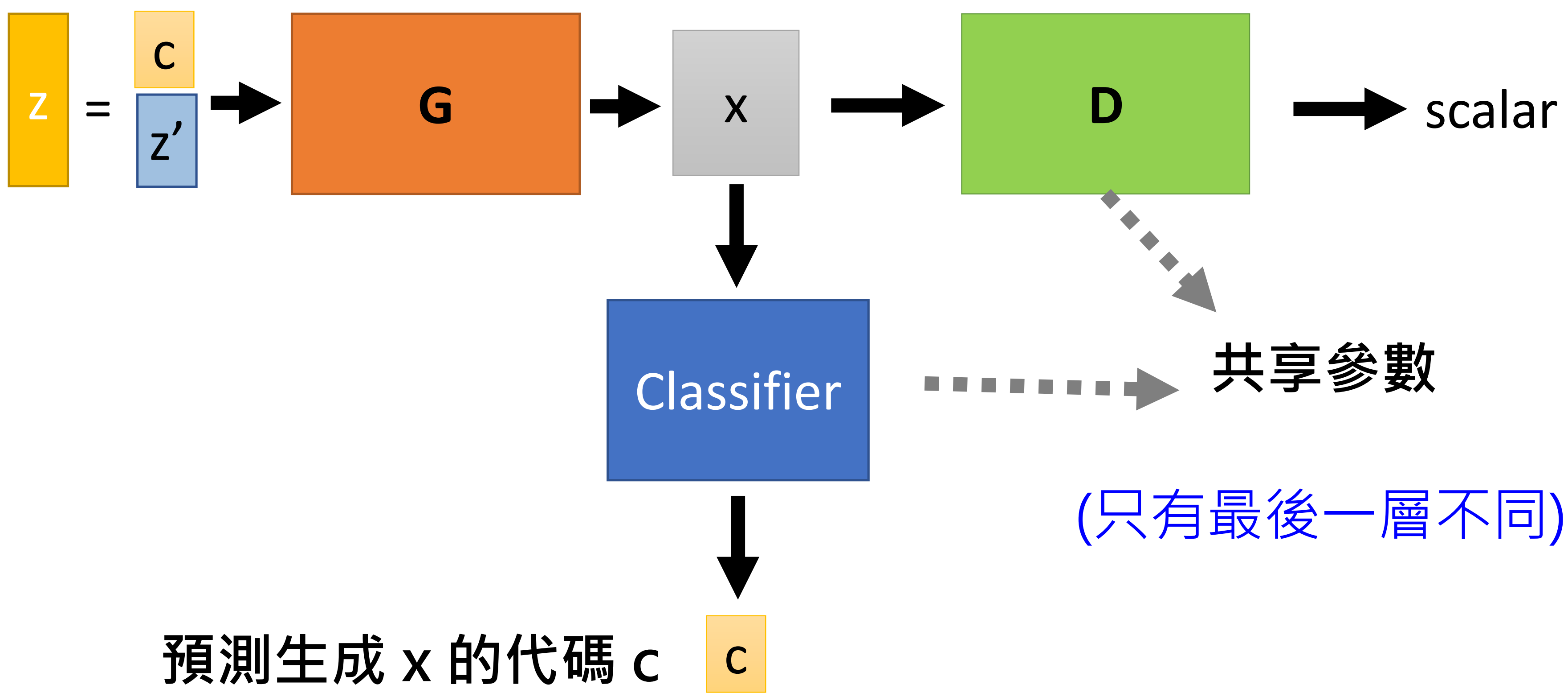
實際



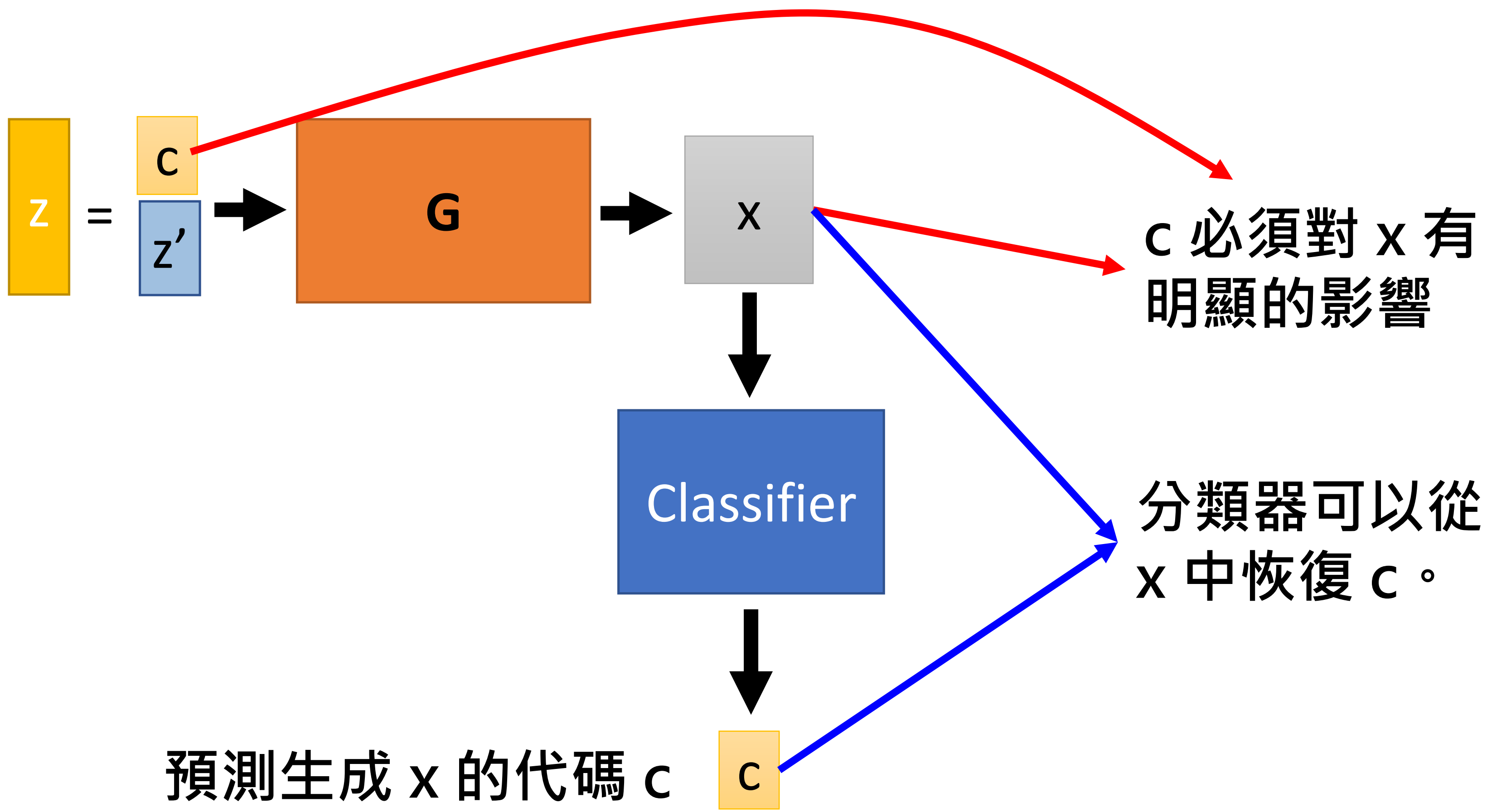
- 顏色代表特徵

[13]

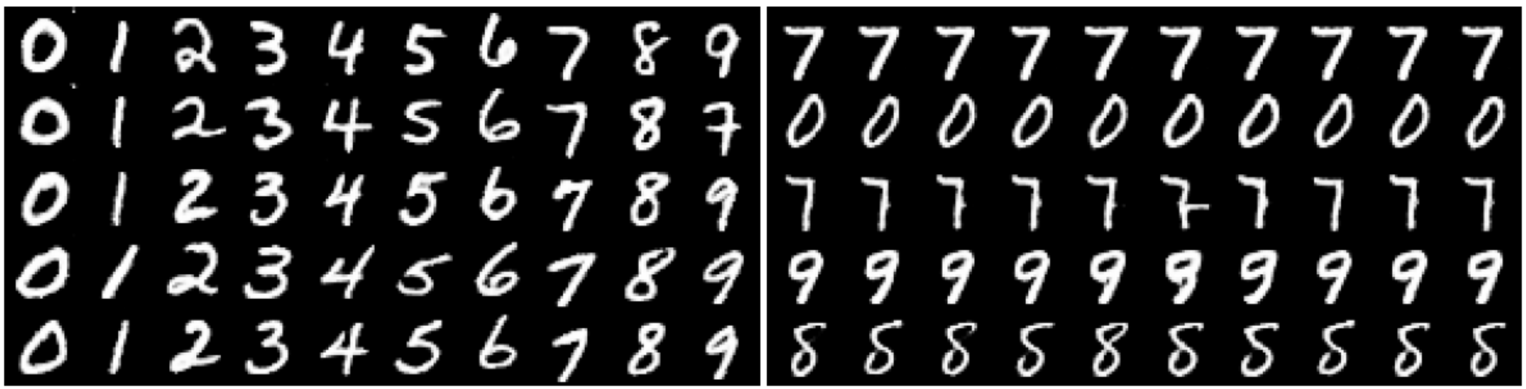
InfoGAN(2/4)



InfoGAN(3/4)

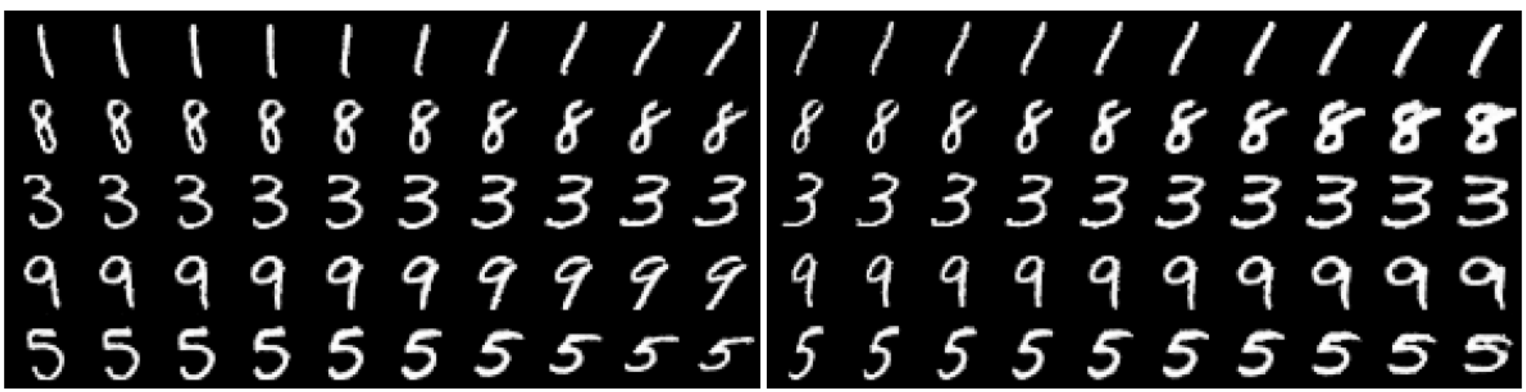


InfoGAN(4/4)



(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)



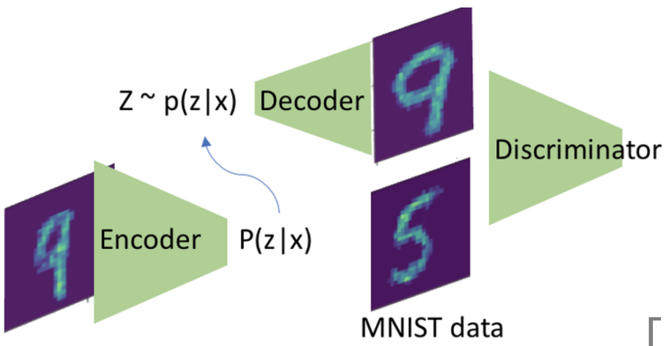
(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

[13]

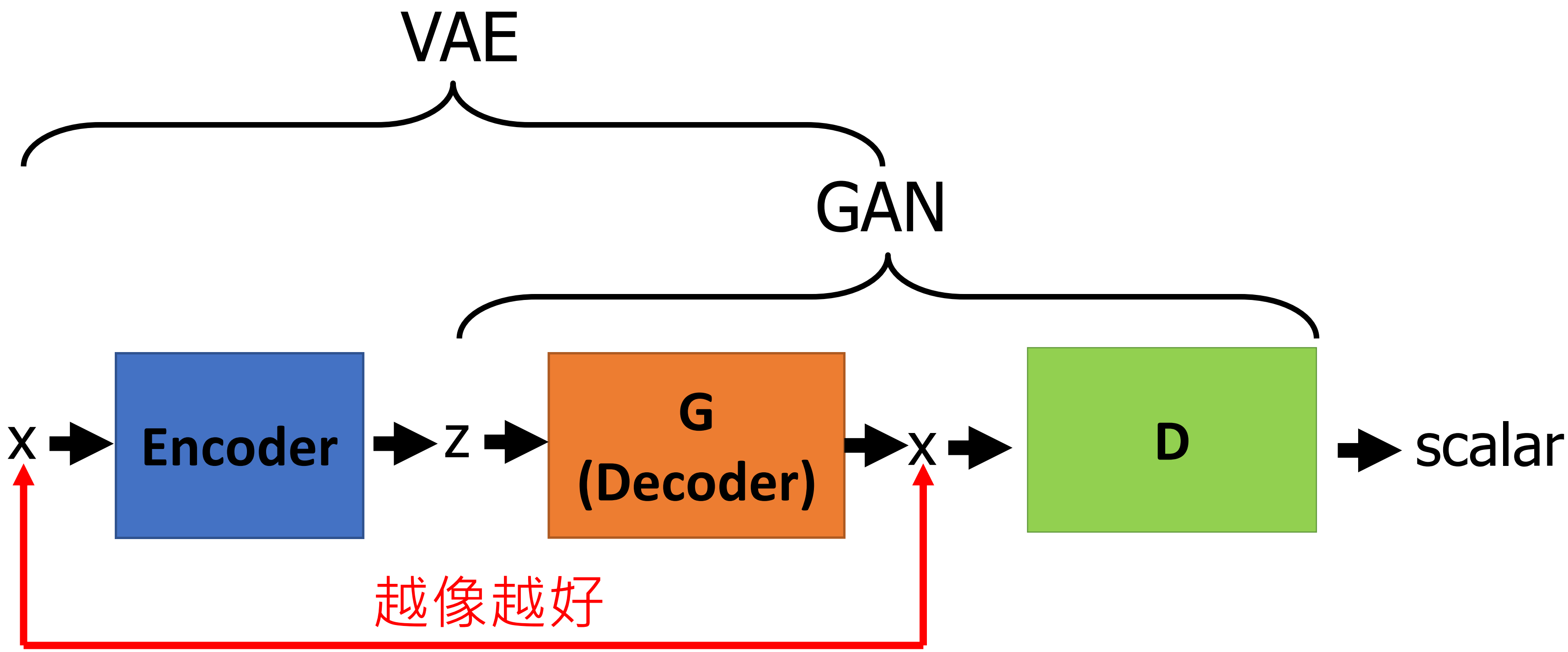
VAE-GAN(1/2)

- VAE-GAN(Variational Auto Encoder Generator)
- 原始GAN 中，透過輸入隨機的向量讓它生成出對應的圖像，但生成器並沒有看過真正的圖像。
- 在訓練過程中，必須花非常多的時間調整參數。
- VAE-GAN 在一開始就見過真正的圖像，因此 VAE-GAN 的訓練會相對穩定不少。



[14]

VAE-GAN (2/2)

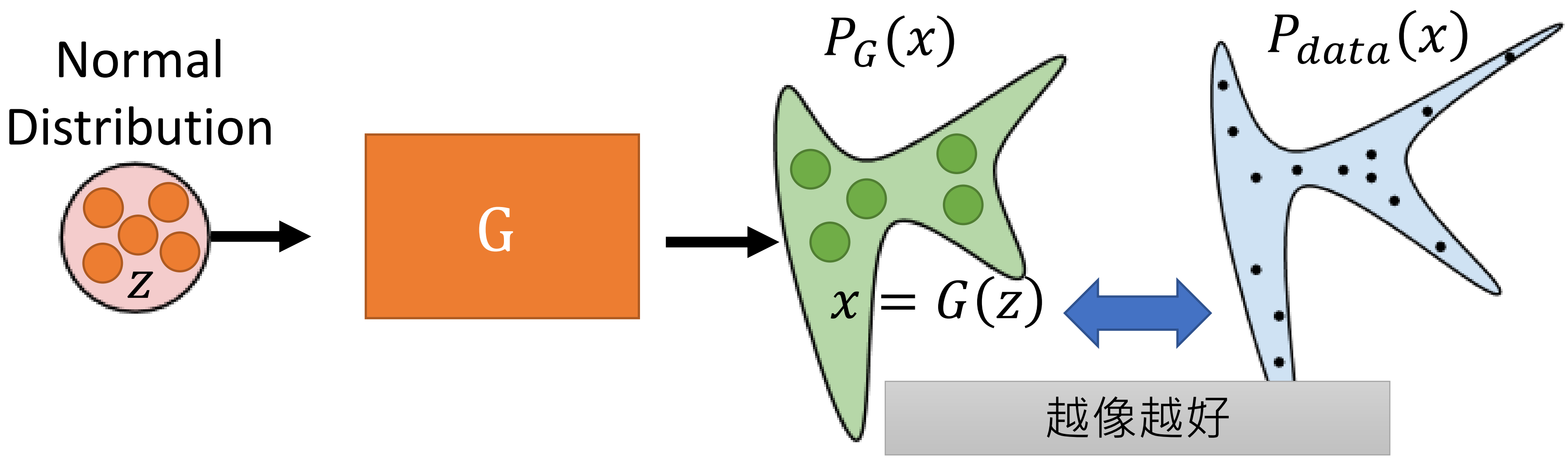


- 最小化重建誤差
- 騙過判別器

- 區分重建的圖像是真實的或是生成

Tips to Implement GAN : Loss Function

GAN



$$G^* = arg \min_G \underline{Div(P_G, P_{data})}$$

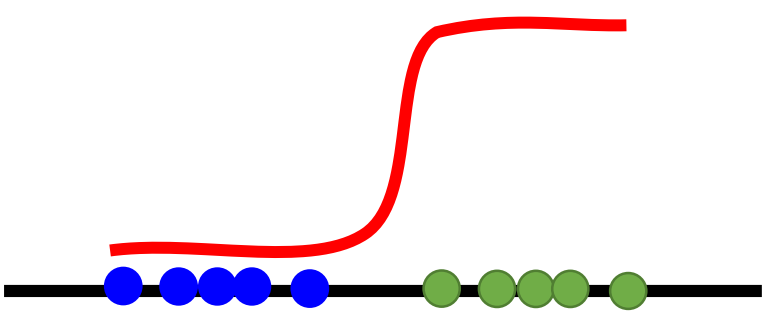
分佈P_G 和 P_{data}之間的差異

LSGAN(Mao,2017)

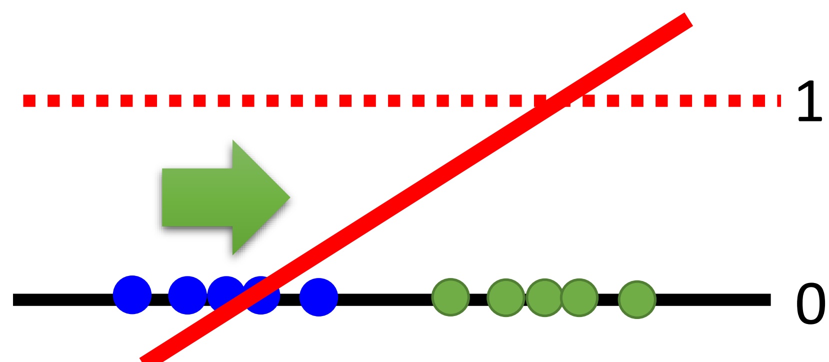
- Least Square GAN (LSGAN)
- 認為原始GAN生成圖片質量不高
- 使用不同的距離度量來構建一個更加穩定而且收斂更快
- 用線性替換 sigmoid

真實影像
生成影像

- sigmoid



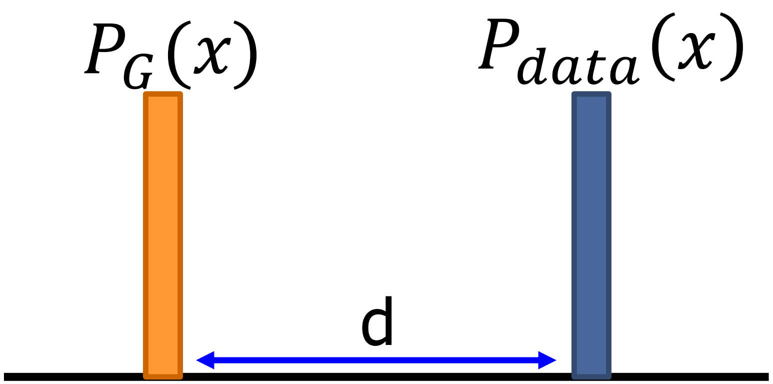
- linear



[11]

WGAN (Arjovsky,2017)(1/2)

- Wasserstein GAN (WGAN): Earth Mover’ s Distance
- 使用不同的距離度量來構建一個更加穩定而且收斂更快
- 考慮一個分佈 P_G 為一堆土，另一個分佈 P_{data} 為目標
- 推土機必須移動地球的平均距離。
- 使用平均距離最小的“移動計劃”來定義推土機的距離。

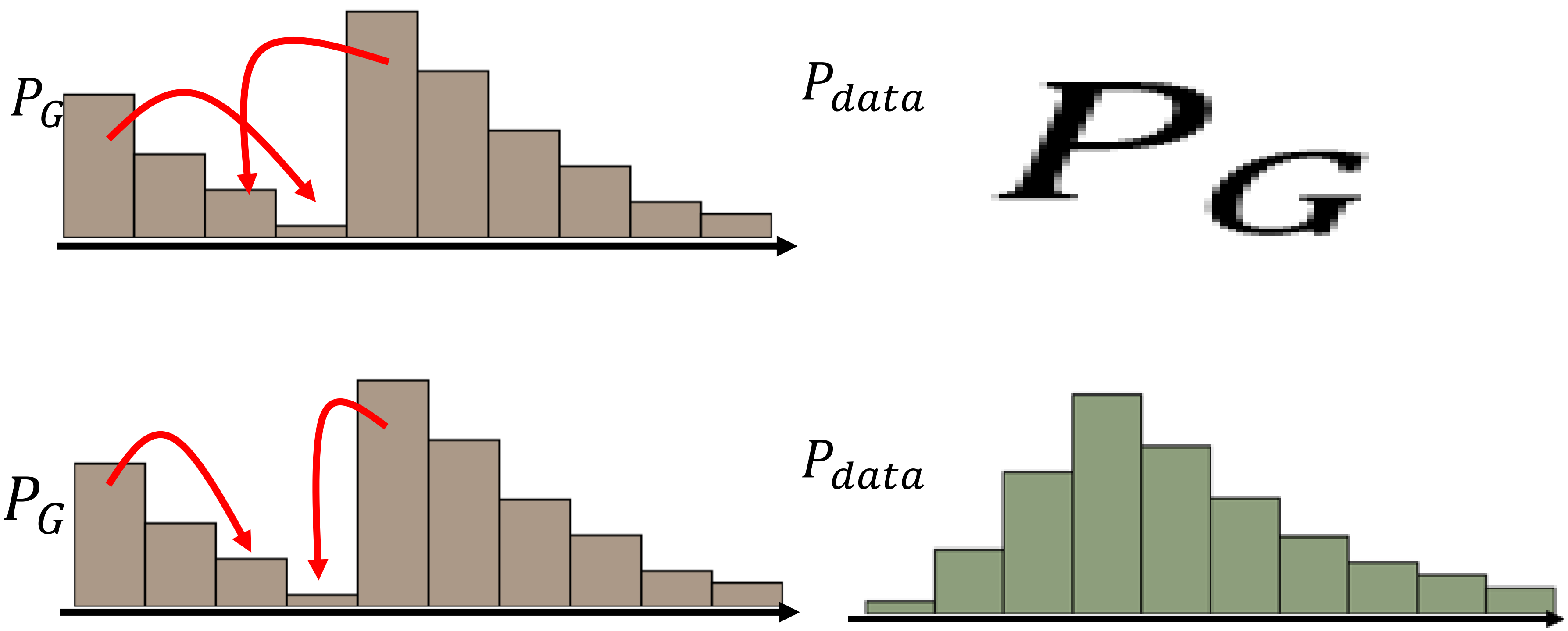


$$W(P_G, P_{data}) = d$$

[12]

WGAN (2/2)

- Wasserstein GAN (WGAN): Earth Mover'　s Distance



比較(1/2)

	GAN	CGAN	StackGAN	CycleGAN	StarGAN
資料類型	大多為圖像	文字、圖像、影音	文字、圖像	圖像、影片	圖像、影片
輸入來源	大多為圖像	• 成對文字及圖像 • 成對圖像	• 成對文字及圖像	兩種類風格圖像	多種類風格圖像
功能	生成圖像	透過條件生成圖像	透過條件生成圖像	圖像轉譯	圖像轉譯
應用	• 資料增強 • 圖像生成	• 資料增強 • 文字轉換圖像 • 圖像合成	• 資料增強 • 文字轉換圖像	• 資料增強 • 風格轉換	• 資料增強 • 風格轉換

比較(2/2)

Feature Extraction

	VAE-GAN	InfoGAN
資料類型	圖像	圖像
輸入來源	大多為圖像	大多為圖像
功能	優化特徵提取	優化特徵提取
應用	• 資料增強 • 圖像生成	• 資料增強 • 圖像生成

Loss function

	LSGAN	WGAN
資料類型	圖像	圖像
輸入來源	大多為圖像	大多為圖像
功能	優化損失函數	優化損失函數
應用	• 資料增強 • 圖像生成	• 資料增強 • 圖像生成

Demo

Import required packages

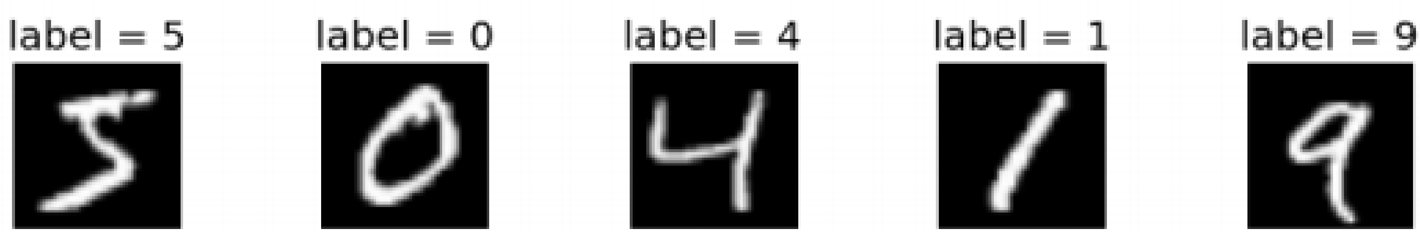
Import

```
[ ]  import time
     import torch
     import torch.nn as nn
     import torch.optim as optim
     from torch.utils.data import DataLoader
     from torchvision import datasets
     from torchvision.transforms import transforms
     import numpy as np
     import matplotlib.pyplot as plt
```

Load data

- Mnist dataset

```
# Load data
train_set = datasets.MNIST('mnist/', train=True, download=True, transform=transform)
test_set = datasets.MNIST('mnist/', train=False, download=True, transform=transform)
train_loader = DataLoader(train_set, batch_size=batch_size, shuffle=True)
test_loader = DataLoader(test_set, batch_size=batch_size, shuffle=False)
```



Build model

- Discriminator

```
class  discriminator(nn.Module):
        def  __init__(self):
                super(discriminator,  self).__init__()
                self.main  =  nn.Sequential(
                        nn.Linear(784,  256),
                        nn.LeakyReLU(0.2),
                        nn.Linear(256,  256),
                        nn.LeakyReLU(0.2),
                        nn.Linear(256,  1),
                        nn.Sigmoid()
                )

        def  forward(self,  input):
                return  self.main(input)
```

Build model

- Generator

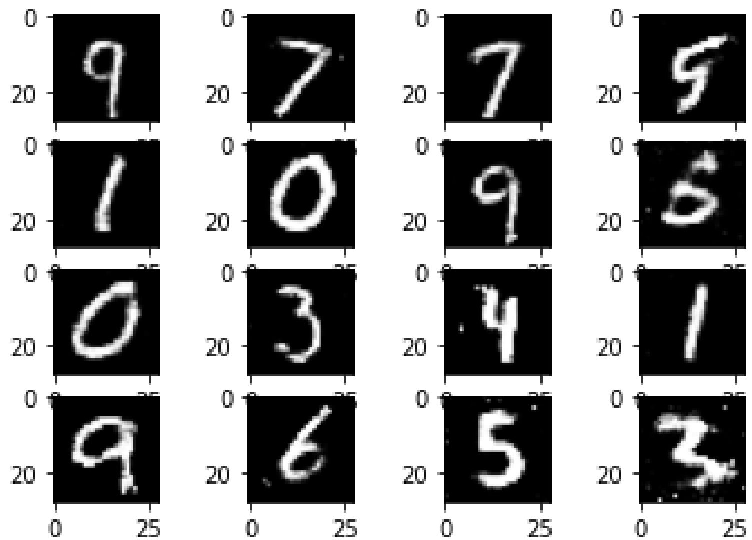
```
class generator(nn.Module):
    def __init__(self):
        super(generator, self).__init__()
        self.main = nn.Sequential(
            nn.Linear(128, 1024),
            nn.ReLU(),
            nn.Linear(1024, 1024),
            nn.ReLU(),
            nn.Linear(1024, 784),
            nn.Tanh()
        )

    def forward(self, input):
        return self.main(input)
```

Show image

```
def show_images(images):
    sqrtn = int(np.ceil(np.sqrt(images.shape[0])))

    for index, image in enumerate(images):
        plt.subplot(sqrtn, sqrtn, index+1)
        plt.imshow(image.reshape(28, 28))
```



Loss function

- Discriminator loss

```
[ ]  # Discriminator Loss => BCELoss
     def d_loss_function(inputs, targets):
         return nn.BCELoss()(inputs, targets)
```

- Generator loss

```
def g_loss_function(inputs):
    targets = torch.ones([inputs.shape[0], 1])
    targets = targets.to(device)
    return nn.BCELoss()(inputs, targets)
```

parameter settings

- Parameter

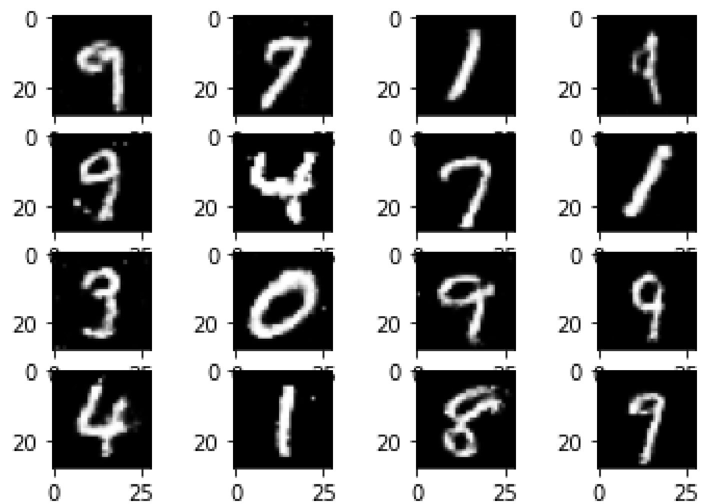
```
#  Settings
epochs  =  200
lr  =  0.0002
batch_size  =  64
g_optimizer  =  optim.Adam(G.parameters(),  lr=lr,  betas=(0.5,  0.999))
d_optimizer  =  optim.Adam(D.parameters(),  lr=lr,  betas=(0.5,  0.999))
```

National Tsing Hua University

INNOVATION ENGINEERING LAB

Show result

```
[200/200, 100/938] D_loss: 0.599 G_loss: 0.845
[200/200, 200/938] D_loss: 0.573 G_loss: 1.244
[200/200, 300/938] D_loss: 0.618 G_loss: 0.913
[200/200, 400/938] D_loss: 0.662 G_loss: 0.955
[200/200, 500/938] D_loss: 0.605 G_loss: 0.764
[200/200, 600/938] D_loss: 0.580 G_loss: 1.048
[200/200, 700/938] D_loss: 0.590 G_loss: 0.890
[200/200, 800/938] D_loss: 0.612 G_loss: 1.281
[200/200, 900/938] D_loss: 0.561 G_loss: 1.021
[200/200, 938/938] D_loss: 0.680 G_loss: 0.715
```



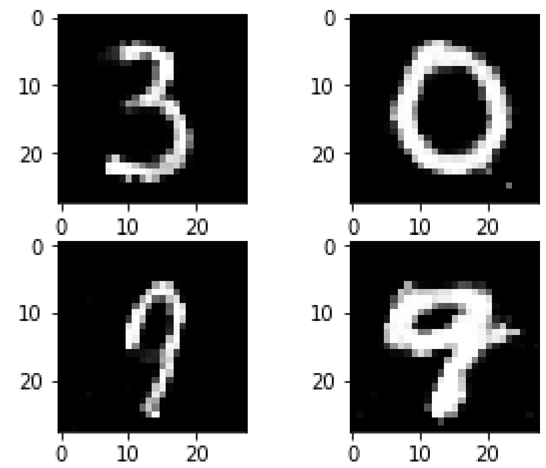
```
Model saved.
Training Finished.
Cost Time: 13996.188408136368s
```

Testing

```
# Model
G = torch.load('Generator_epoch_200.pth')
G.eval()

# Generator
noise = (torch.rand(16, 128)-0.5) / 0.5
noise = noise.to(device)

fake_image = G(noise)
imgs_numpy = (fake_image.data.cpu().numpy()+1.0)/2.0
show_images(imgs_numpy)
plt.show()
```



Class Assignment & Homework

Class assignment

- 請使用助教提供的Mnist 數據集預訓練權重，來使GAN（對抗神經網路）來生成數字圖像
- 以ipynb格式提交您的作業，並另外將測試結果的圖檔以png格式上傳。

Homework

- 請使用MNIST資料集和參考助教提供的GAN程式，訓練一個GAN模型
- 使用以下超參數設置

Epochs	Learning rate	Batch_size
300	0.0002	32

- 以ipynb格式提交您的作業，並另外將測試結果的圖檔以png格式上傳。

Reference

GAN

- [1] https://ithelp.ithome.com.tw/articles/10196257
- [2] Generative adversarial networks: a survey on applications and challenges
- [3] https://speech.ee.ntu.edu.tw/~tlkagk/courses_MLDS18.html

CGAN

- [4] Generative Adversarial Text to Image Synthesis
- https://blog.csdn.net/qq_24224067/article/details/104293409
- https://zhuanlan.zhihu.com/p/35983991
- [6] Image-to-Image Translation with Conditional Adversarial Networks

Reference

Stack GAN

- [5] StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks

Cycle GAN

- [7] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- https://zhuanlan.zhihu.com/p/45394148
- https://www.gushiciku.cn/pl/gkTn/zh-tw

Reference

Disco GAN & Dual GAN

- [8] Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- [9] DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- https://zhuanlan.zhihu.com/p/44562727

StarGAN

- [10] StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

Reference

LSGAN

- [11] Least Squares Generative Adversarial Networks
- https://zhuanlan.zhihu.com/p/25768099

WGAN

- [12]Wasserstein GAN
- https://medium.com/@falconives/day-59-wasserstein-gan-wgan-b31adb226aea

InfoGAN

- [13] InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- https://zhuanlan.zhihu.com/p/58261928

Reference

VAEGAN

- [14]Autoencoding beyond pixels using a learned similarity metric
- https://blog.csdn.net/a312863063/article/details/83576624

SRGAN

- [15]Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Image completion

- [16]Globally and Locally Consistent Image Completion