



Small Data GAN



友達未來學院

陽明交大 AI 學院 魏澤人
tjw@nctu.edu.tw

About Me

交大 AI 學院/緯創資通 AI 青年講座

Google Developers Expert (Machine Learning)

Organizer, GDG Hualien, Hualien.py, ...

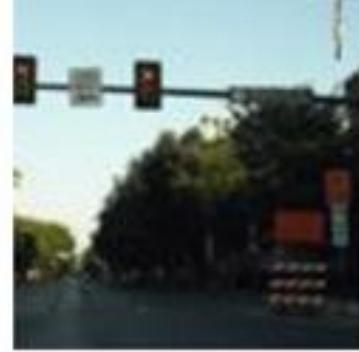
Chief Mathematician, iiNumbers

AI and Math Consultant, Ubitus, ITRI, ...

AWS Cloud Ambassador



生成模型



給定一些例子



$G(z_1) =$  $G(z_2) =$  $G(z_3) =$ 













鶴調前陵讌嬖藐僂豎蹤搦襖鄆滄懃府
滌咀麌筭芹琬度漁梢顛什僂銢絨莫干
翠唾墮韞縵磬爻心櫟櫬鼻垂憐睇顆闇摺
垂譚甡娼羃駢丙倫嶠齧哂馨祁庭騰僂
圖暎面苑傖脰郿鄧抿痈為疥鱠嶙櫈遑鳴
躊躋嫋諉甕颶坫汙濶輒貌橐迺旋嫗弔觀
飄狗熾鴻保汎之礪搘死鞚窈珥扳壘慘
獨曠博因岷啟蹙猝潾膩辯慝艣殉懃角
桷刁鞚繫佞性瞷鱸適鮒卽迺黷敝媾汨櫟
星冠造漆逛鶴僻疋眭嚼睽侈脣險尙崢
冠躋羣鑿鋰俟痵渝有寘露痂釀忽峙挂
沽鏤磁您蔡兀攜肥愣擾懼卷猷鎗鉦叵櫟
鈎踞棘纂鉢攀蠭珊瑚炸瘰錙疣奕墳椒
煥屬睨寫葱蘢瀝癥啾利靄坐口鈎法鄂
灣駢敵操恆顛榦魄壞融寇佛繫鯁對踽

津脩儀雜曩矯頓悶饅慘病
眇灘嬪刻瑢蔚侍麌倬志涯返姽幃櫛櫳糴
肺癉湫玗鬼耆鑪鑿頽滯烽券屬謊涅兒
賂涕喻檄伯沃惔紜枳韓獻紛泊芸瓦抹
賺炙杷爬儈眞踰爻轎遫魘肸併婍趨魏
柵梧鍼恍批俠柄樘瞠檻橙揔賴遷媯佇
嵌雁轆纏拉暑庳標豬誕鉛喚岸嬪跗
箋魄校猖摻騶蟋蹠賞礎暱驪嘒蓑嬖砬
巖蛙鬢翦舒約離澤黔趁蚨桔橙躋暖氛
翬漳沚頓鵠僞柏氤鷄雲蹕宛墻亞恚彷
諤悠馳薑輓勒鷁剜爛嫪狼柒榦椀啜瘁
呣藏粢蟲號斲蹀胚茹劫獯冕芟拂銙飄
嫜磨鞬侗懲汰蠟擎稈筭菩懋荆攜褶屹
焦柵祿恨厥汨濺闔頽蠢貞籬続鯔瀛臚
澁璇啖忘唱瘡錢朵璫蟹置戒郎鮒實酥
鍾跔痘泣鱠摹驥嵒綺蕪籬跨獲裘弦



TEXT PROMPT

a pentagonal brown clock. a brown clock in the shape of a pentagon.

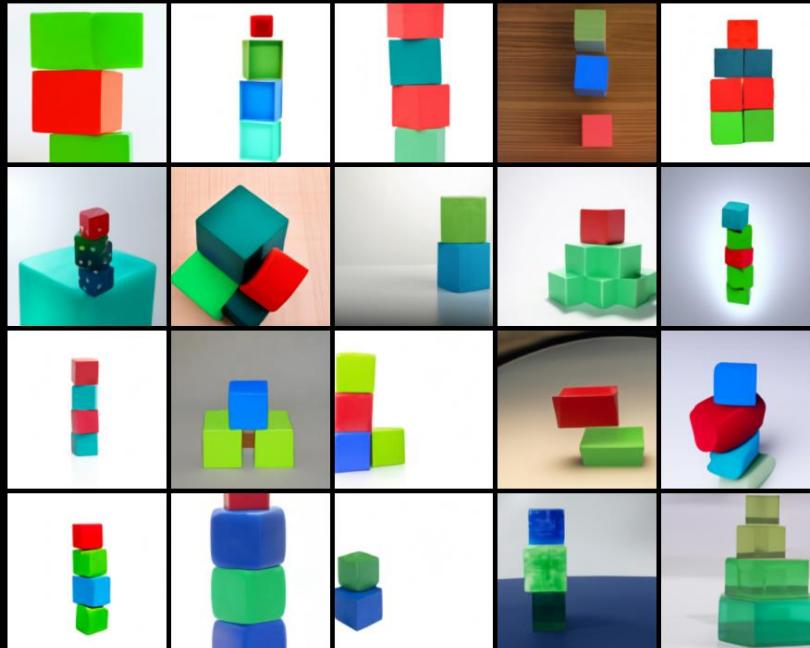
AI-GENERATED IMAGES

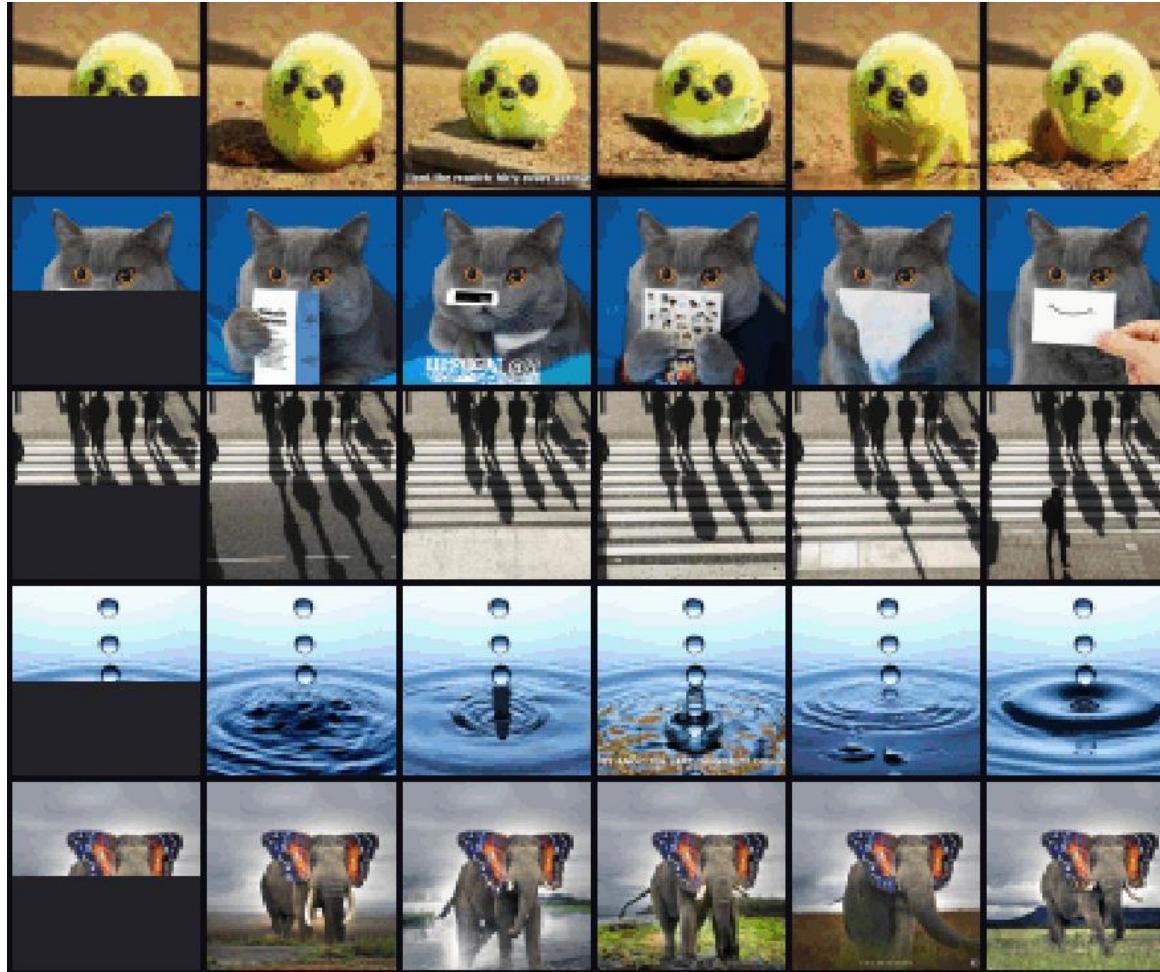


TEXT PROMPT

a stack of 3 cubes. a red cube is on the top, sitting on a green cube. the green cube is in the middle, sitting on a blue cube. the blue cube is on the bottom.

AI-GENERATED IMAGES





前面都
不是用 *GAN*

VQVAE2

GLOW

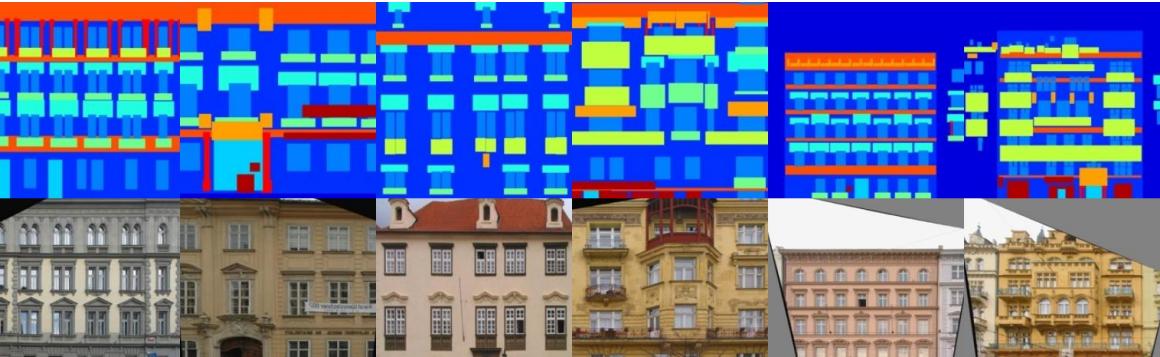
Diffusion Model

Image GPT

DALI

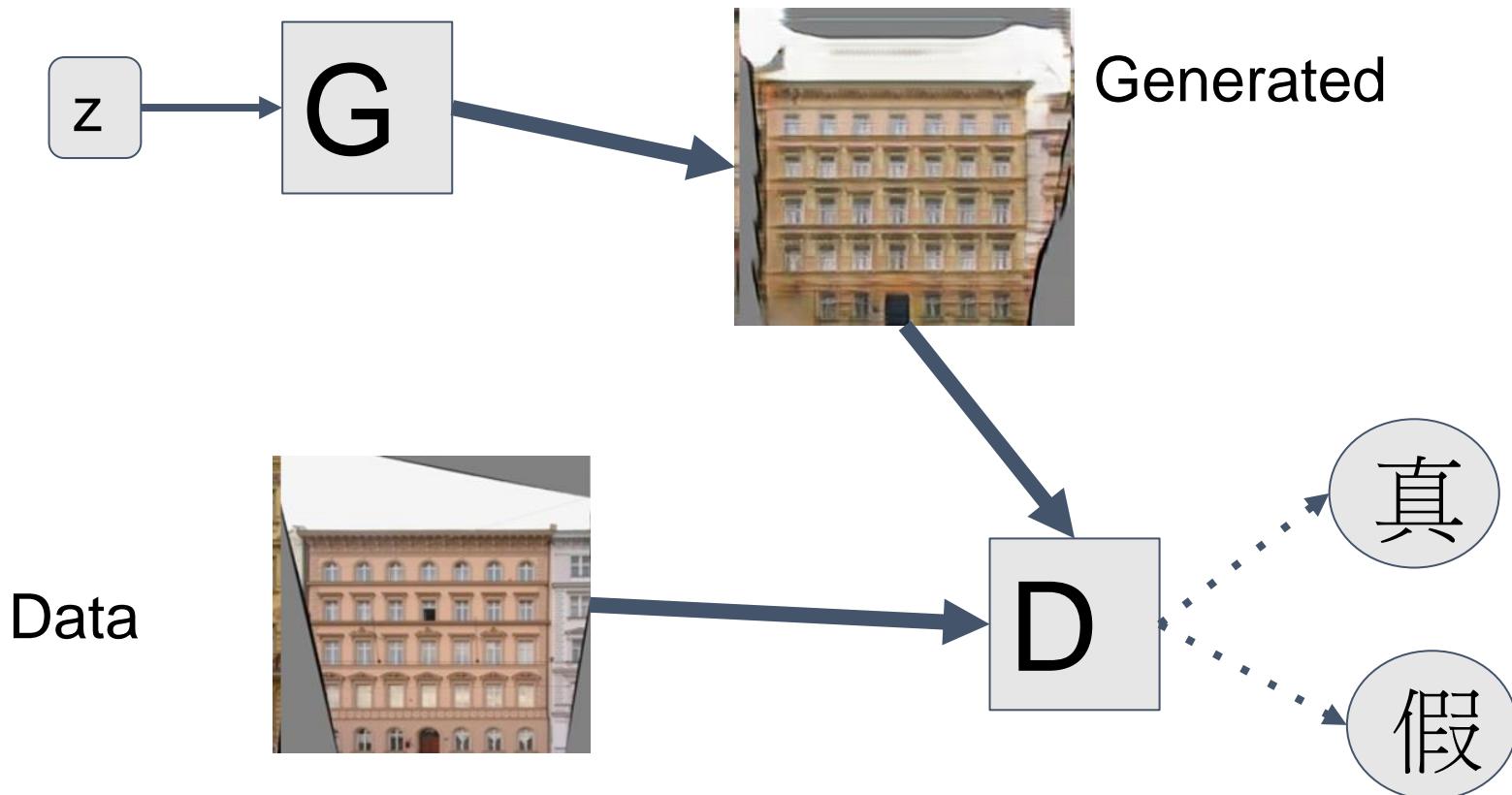
A photograph of two men from the chest up. They are both wearing dark-colored, ribbed knit beanies. The man on the left has a yellow pom-pom on his beanie and is looking directly at the camera with a slight smile. The man on the right has a pinkish-red beanie and is also looking towards the camera. The background is dark and out of focus.

生成對抗 網路介紹





Generator vs Discriminator





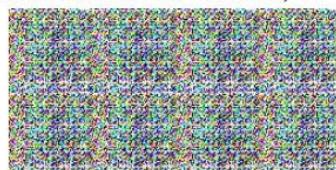
DCGAN |

DCGAN**LSGAN****WGAN (clipping)****WGAN-GP (ours)**

Baseline (G : DCGAN, D : DCGAN)



G : No BN and a constant number of filters, D : DCGAN

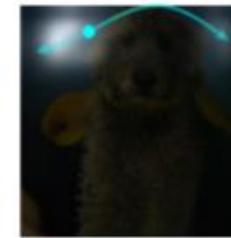
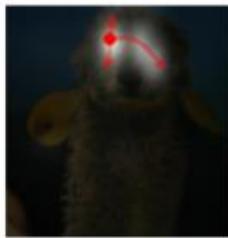
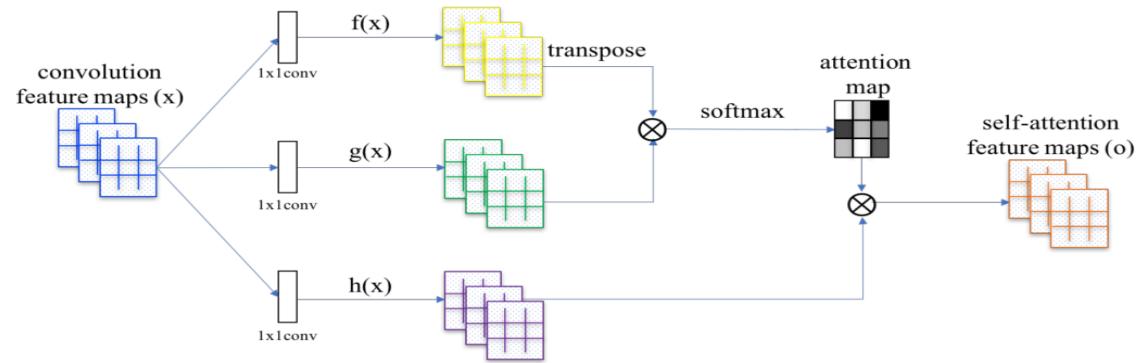


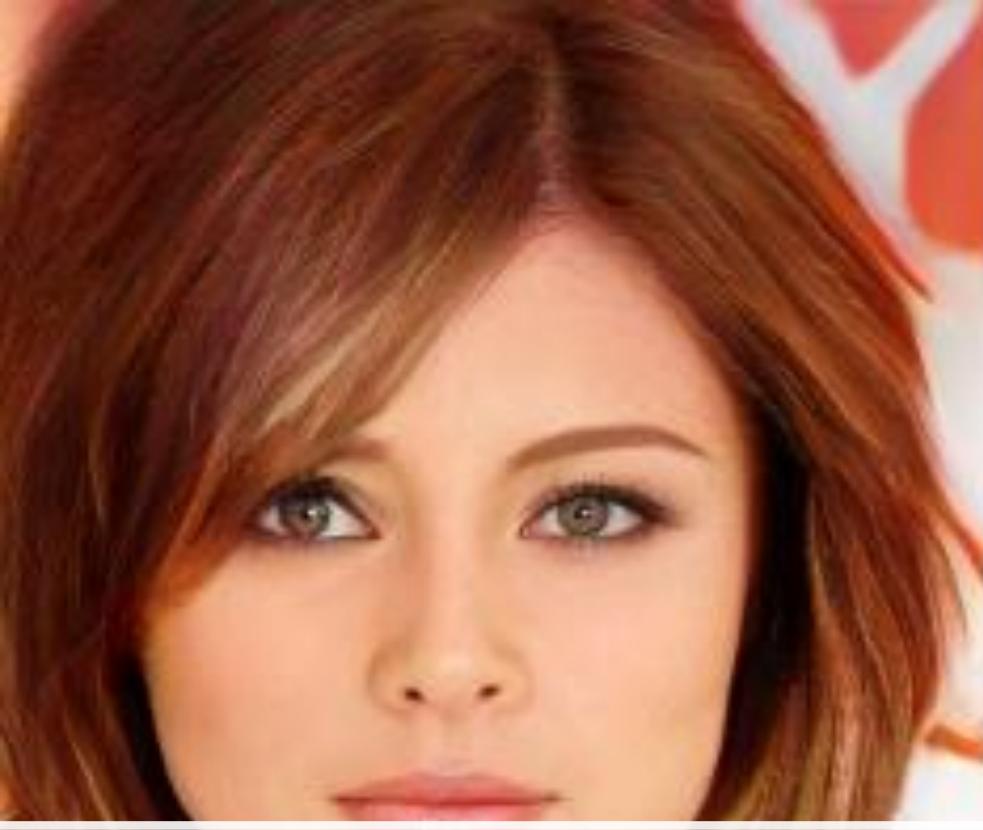
G : 4-layer 512-dim ReLU MLP, D : DCGAN



No normalization in either G or D







Progressive Growing of GANs



StyleGAN

StyleGAN2



Alias-Free GAN (Ours)



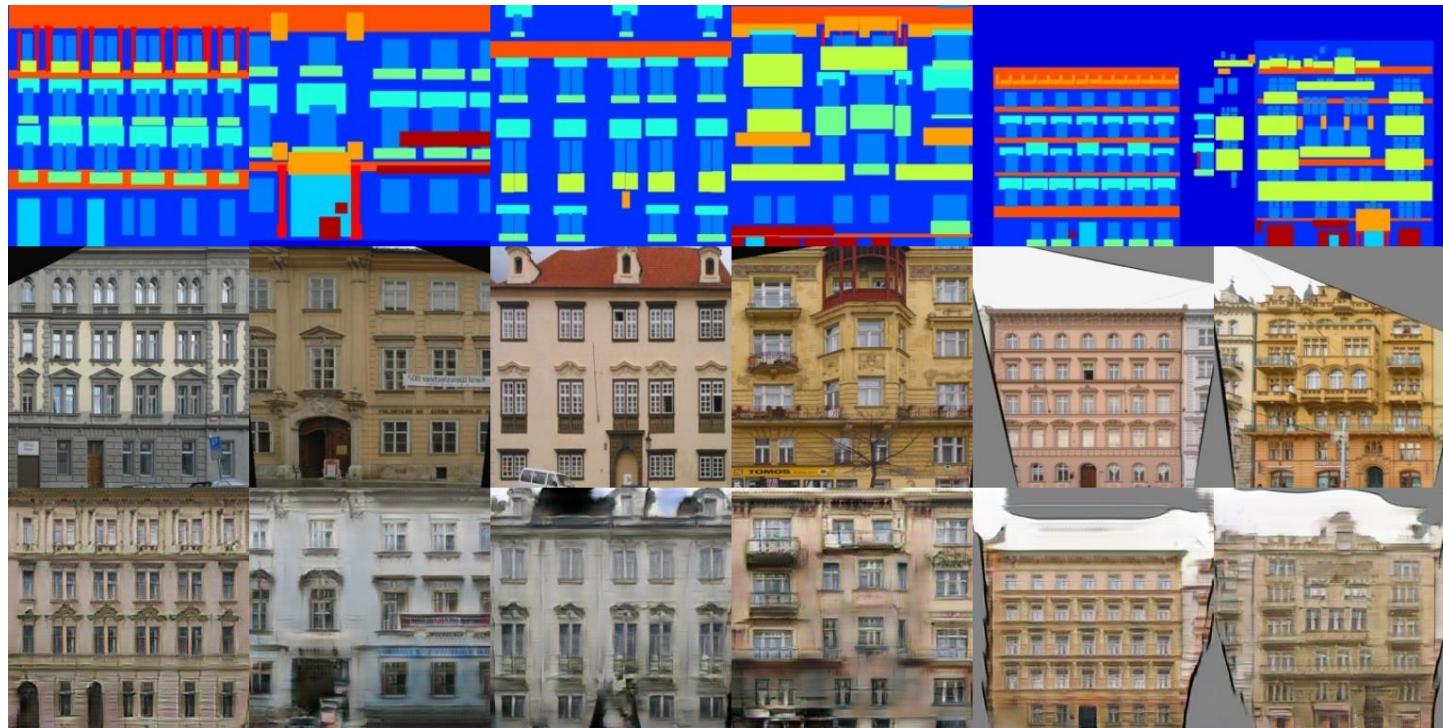
Random latent walk using directions from StyleCLIP, GANSpace, and SeFa.

Pix2Pix



Image-to-Image Translation with Conditional Adversarial Networks

[Phillip Isola](#), [Jun-Yan Zhu](#), [Tinghui Zhou](#), [Alexei A. Efros](#)



CycleGAN



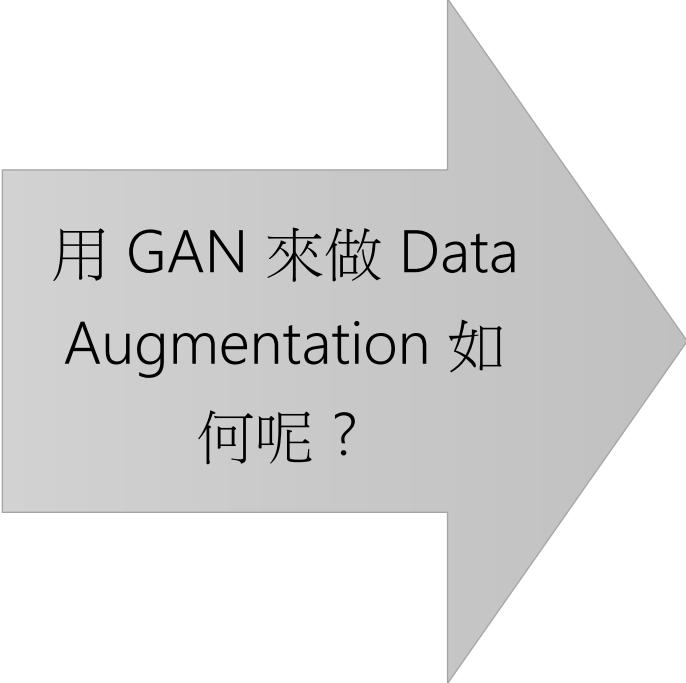
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu*, Taesung Park*, Phillip Isola, Alexei A. Efros Berkeley AI Research Lab, UC Berkeley



Data Augmentation Using GAN

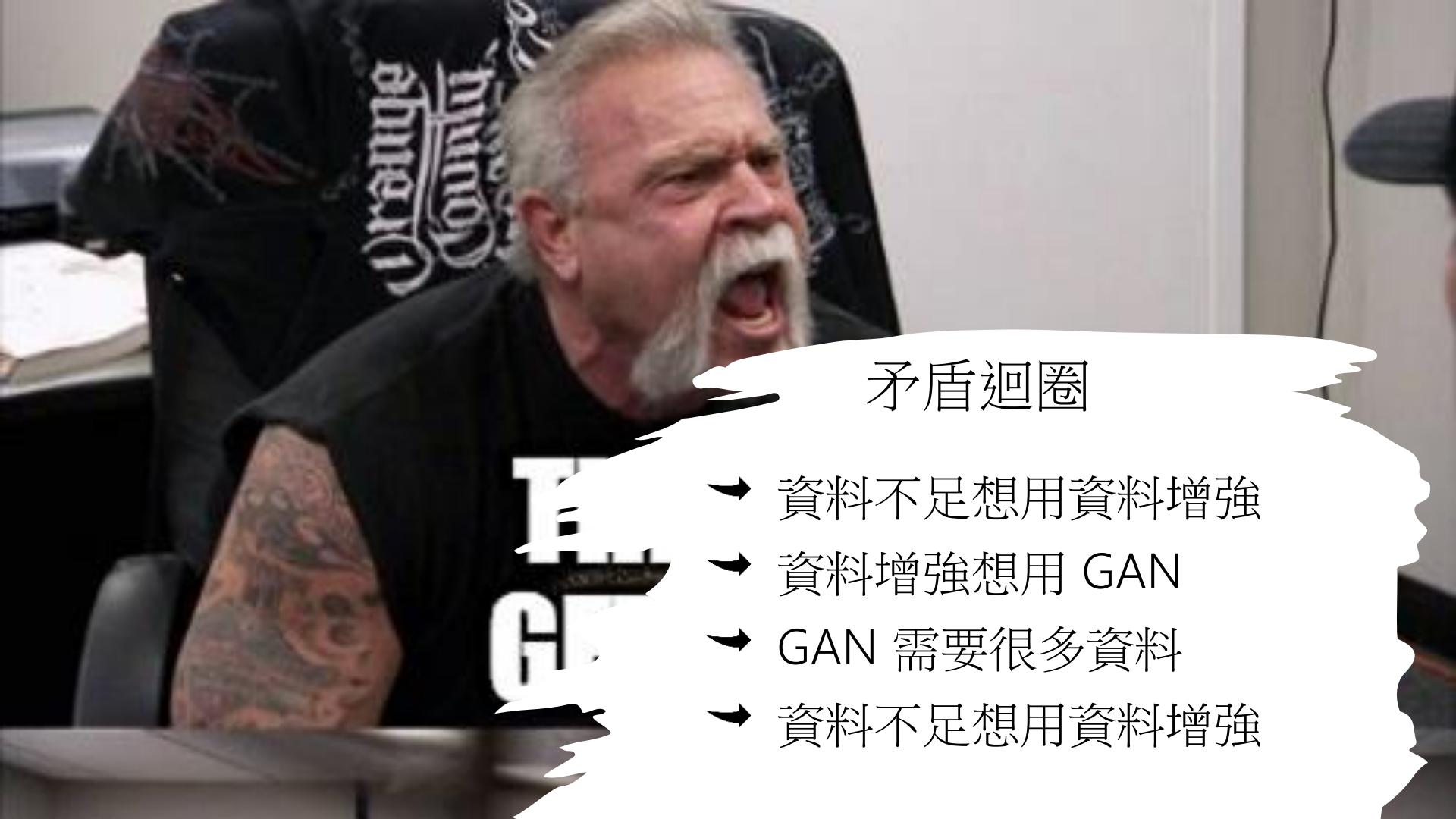
Data Augmentation Using GAN



用 GAN 來做 Data
Augmentation 如
何呢？



情境：資料不足、
資料不平衡



矛盾迴圈

- 資料不足想用資料增強
- 資料增強想用 GAN
- GAN 需要很多資料
- 資料不足想用資料增強

數值資料的 資料增強

<https://paperswithcode.com/paper/dataset-augmentation-using-gans>

SMOTE

- Synthetic Minority Oversampling Technique
 - <https://www.jair.org/index.php/jair/article/view/11192/26406>
 - 2000 年 , Chawla 在研究所 ,
 - 要用 tree classifier 做一個關於乳癌資料的分類問題。
 - 本來覺得不容易
 - 但 decision tree 就達到準確度 97%。
 - 有一瞬間他很高興
 - 很快就發現一個嚴重的問題 , 那就是資料裡面有 97.68% 是同一個類別的。
 - 有超過 80 個 SMOTE 變形

原理

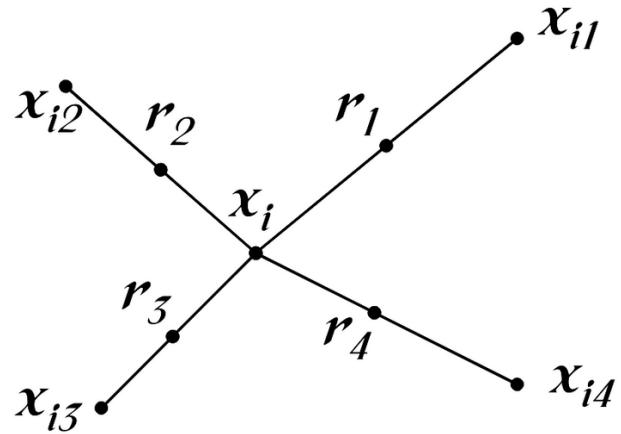


Figure 1: An illustration of how to create the synthetic data points in the SMOTE algorithm

Algorithm 1 SMOTE algorithm

```
1: function SMOTE( $T, N, k$ )
   Input:  $T$ ;  $N$ ;  $k$             $\triangleright$  #minority class examples, Amount of oversampling, #nearest
   neighbors
   Output:  $(N/100) * T$  synthetic minority class samples
   Variables:  $Sample[][]$ : array for original minority class samples;
    $newindex$ : keeps a count of number of synthetic samples generated, initialized to 0;
    $Synthetic[][]$ : array for synthetic samples
2:   if  $N < 100$  then
3:     Randomize the  $T$  minority class samples
4:      $T = (N/100)*T$ 
5:      $N = 100$ 
6:   end if
7:    $N = (\text{int})N/100$   $\triangleright$  The amount of SMOTE is assumed to be in integral multiples
   of 100.
8:   for  $i = 1$  to  $T$  do
9:     Compute  $k$  nearest neighbors for  $i$ , and save the indices in the  $nnarray$ 
10:    POPULATE( $N, i, nnarray$ )
11:   end for
12: end function
```

Algorithm 2 Function to generate synthetic samples

```
1: function POPULATE( $N, i, nnarray$ )
   Input:  $N; i; nnarray$   $\triangleright$  #instances to create, original sample index, array of nearest
   neighbors
   Output:  $N$  new synthetic samples in Synthetic array
2:   while  $N \neq 0$  do
3:      $nn = \text{random}(1, k)$ 
4:     for attr = 1 to numattrs do  $\triangleright numattrs =$  Number of attributes
5:       Compute:  $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$ 
6:       Compute:  $gap = \text{random}(0, 1)$ 
7:        $Synthetic[newindex][attr] = Sample[i][attr] + gap \cdot dif$ 
8:     end for
9:      $newindex++$ 
10:     $N--$ 
11:  end while
12: end function
```

ADASYN

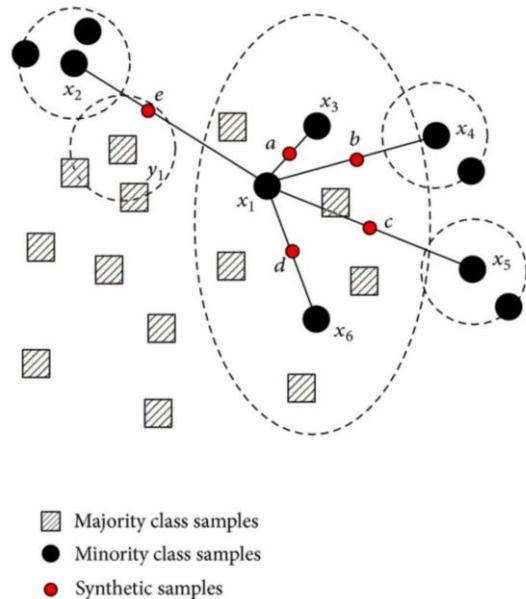


Figure 3: Example of SMOTE (from [Hu and Li \(2013\)](#))

ADASYN

- Compute $r_i = \Delta_i / K$, where, Δ_i is the number of examples in K nearest neighbours of x_i that belong to the majority class.
- Generate $G \frac{r_i}{\sum r_j}$ synthetic data examples for x_i , where G is the total number of synthetic data examples.
- Add random noise

Experiment Setting: Models

- GAN: Simple neural networks
- Classifier: Decision Tree

Data set name	Architecture of the GAN
original data	The first 70% of the original database
256/512/1024	Generated by a GAN with 3 hidden layers with size 256, 512 and 1024
256/512	Generated by a GAN with 2 hidden layers with size 256 and 512
256	Generated by a GAN with 1 hidden layer with size 256
128/256/512	Generated by a GAN with 3 hidden layers with size 128, 256 and 512
128/256	Generated by a GAN with 2 hidden layers with size 128 and 256
128	Generated by a GAN with 1 hidden layer with size 128

Experiment Setting: Datasets

Aa Database Name	# Number of ...	# Size	≡ Label Distribution
Pima Indians Diabetes Database	9	768	No diabetes: 500 , Diabetes: 268
Breast Cancer Wisconsin (Diagnostic) Data Set	32	569	Benign: 357 , Malignant: 212
Credit Card Fraud Detection	31	284807	Non-frauds: 284315 , Frauds: 492

Experiment Results: Distribution Similarity

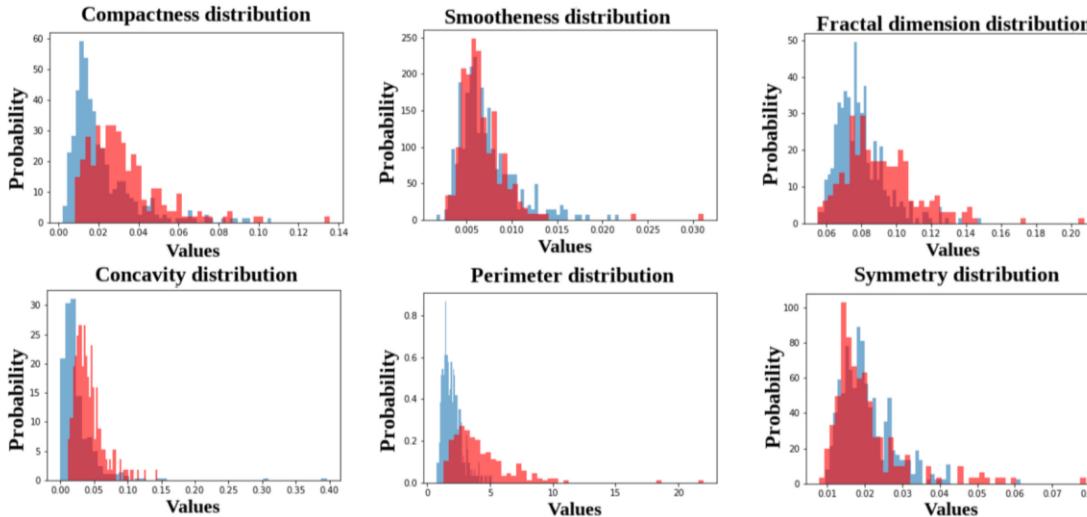


Figure 4: Comparison between the distribution of some attributes in the original (blue) and synthetic (red) cancer data set

Result: Cancer Dataset

Database	Accuracy	Precision	Recall
Original data	0.888	0.679	0.974
256/512/1024	0.91 (0.042)	0.786 (0.127)	0.888 (0.054)
256/512	0.935 (0.03)	0.853 (0.107)	0.896 (0.086)
256	0.907 (0.053)	0.772 (0.126)	0.904 (0.071)
128/256/512	0.869 (0.066)	0.702 (0.144)	0.821 (0.148)
128/256	0.896 (0.048)	0.74 (0.122)	0.908 (0.086)
128	0.906 (0.054)	0.775 (0.131)	0.894 (0.055)

Table 5: Mean and standard deviation ($n=20$) of the classification results in the cancer database.

Result: Diabetes Dataset

Database	Accuracy	Precision	Recall
Original data	0.748	0.784	0.367
256/512/1024	0.682 (0.064)	0.545 (0.093)	0.534 (0.206)
256/512	0.706 (0.05)	0.582 (0.078)	0.584 (0.097)
256	0.601 (0.097)	0.438 (0.118)	0.438 (0.213)
128/256/512	0.685 (0.058)	0.568 (0.109)	0.544 (0.158)
128/256	0.639 (0.094)	0.507 (0.106)	0.579 (0.185)
128	0.653 (0.086)	0.51 (0.117)	0.462 (0.219)

Table 6: Mean and standard deviation (n=20) of the classification results in the diabetes database.

Result: Fraud Dataset (Balanced Accuracy)

Database	Accuracy	Precision	Recall
Original	0.782	1.0	0.565
SMOTE	0.912	0.959	0.861
ADASYN	0.921	0.979	0.861
128	0.807 (0.165)	0.89 (0.202)	0.806 (0.042)
256	0.894 (0.01)	0.998 (0.005)	0.789 (0.018)
128/256	0.902 (0.012)	0.981 (0.015)	0.82 (0.028)
256/512	0.888 (0.032)	0.962 (0.018)	0.808 (0.069)

Summary

- GAN is not a silver bullet
- It may or may not improve the results.
- There are lots of ways to do augmentation
 - Not always better than GAN, but usually easier.



Image Data?

Typically already have some kind of augmentation when training.

SYNTHETIC DATA AUGMENTATION USING GAN FOR IMPROVED LIVER LESION CLASSIFICATION

- Frid-Adar, Maayan, et al. "Synthetic data augmentation using GAN for improved liver lesion classification." 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). IEEE, 2018.

- 探討利用 GAN 來合成 liver lesion 的影像資料集。
- 資料集中原本有 182 張 CT 影像。他們以 lesion 為中心切出方格圖片。
- 傳統的影像增強法混和了旋轉、翻轉、平移及縮放。
- 他們利用傳統增強後的圖片，來訓練 GAN，然後再使用傳統增強。
- 結果：
 - 傳統資料增強，Sensitivity 為 78.6%，Specificity 為 88.4%.
 - 他們的方法，Sensitivity 為 85.7%，Specificity 為 92.4%.

Dataset

3 lesion classes:

- 53 cysts
- 64 metastases
- 65 hemangiomas.

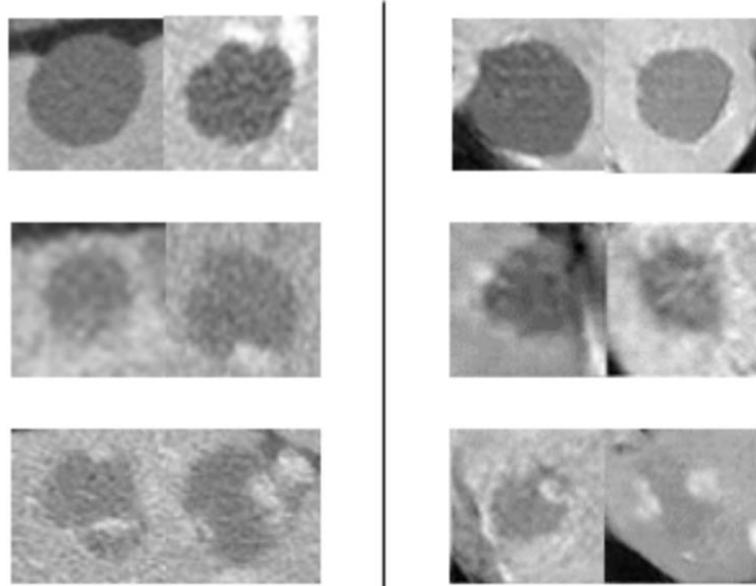


Fig. 1: Lesion ROI examples of Cysts (top row), Metastases (middle row) and Hemangiomas (bottom row). Left side: Real lesions; Right side: Synthetic lesions.

Classic Data Augmentation

最常見的是 Affine transformations

- translation, rotation, scaling, flipping, shearing

Deformation

Perspective transformation

顏色、亮度

雜訊

In this application

- 不是每種都能用，為了確保 liver lesion 的特性不被破壞，不能改變形狀（像是 shearing, elastic deformation ）
- 為了辨識方便，lesion 在 ROI 中心。

The classic augmentations used

- 每個 sample 隨機旋轉 N_{rot} 次，每個再隨機
 - 翻轉 N_{flip} 次（上下、左右）1
 - 平移 N_{trans} 次 ($0.01 \times d$ 的偏移， d 是 lesion 的直徑)
- + :::
 - 縮放 N_{scale} 次
 - 所以總共有 $N = N_{rot} \times (1 + N_{flip} + N_{trans} + N_{scale})$ 個 samples.
 - 最後 resize 成 64×64 .

GAN Model

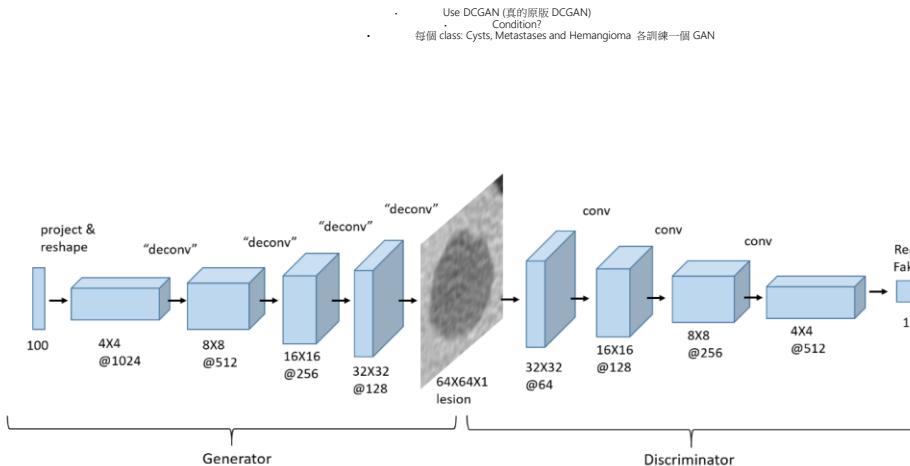
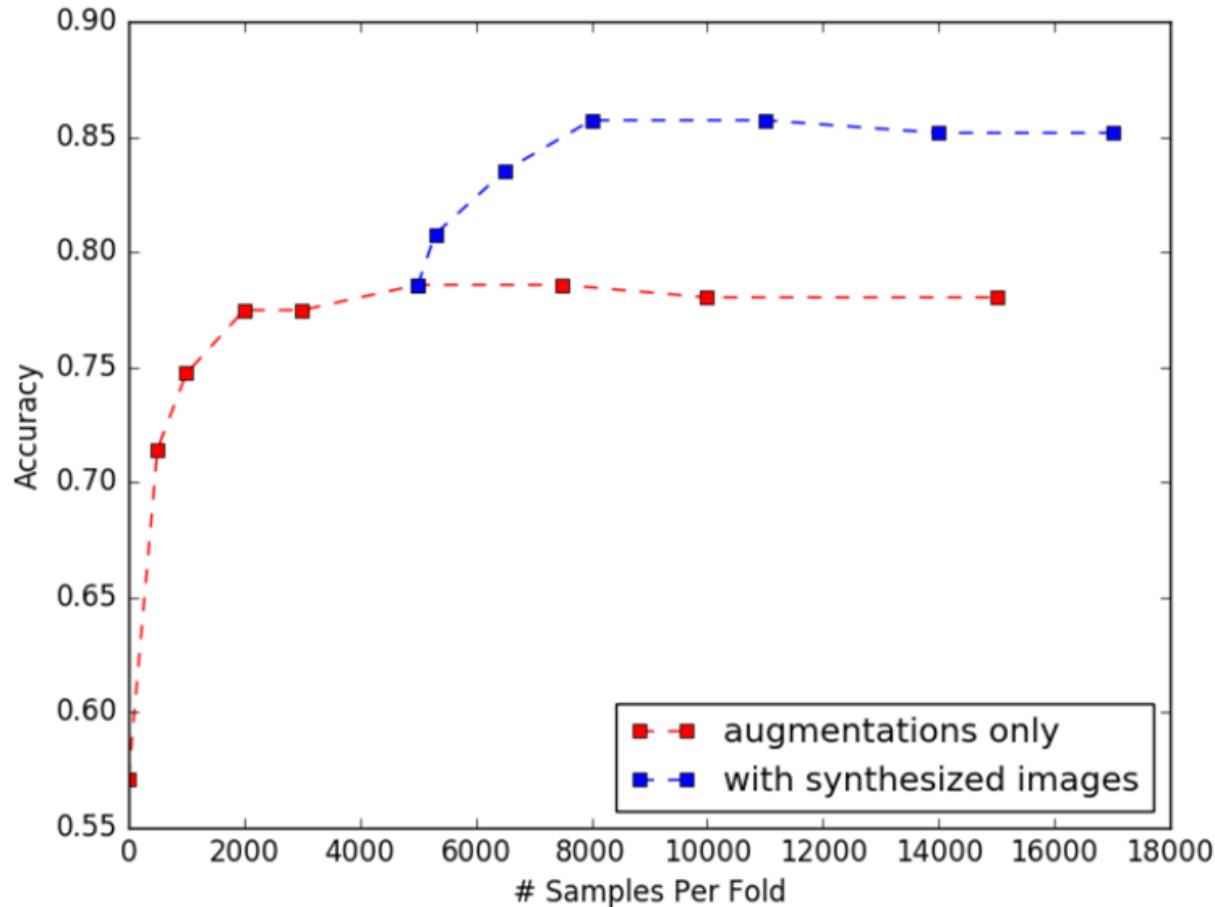


Fig. 2: Deep Convolutional GAN Architecture (generator+descriiminator).

Experiment Setting

- $N = 480$ ($N_{rot} = 30, N_{flip} = 3, N_{trans} = 7, N_{scale} = 5$)
 - Use a not very deep CNN(3 layers, max pooling, ReLU) with dropout
 - 3 lesion classes: 53 cysts, 64 metastases, 65 hemangiomas.
 - 3 fold cross validation, with case separation at the patient level



Classic

True \ Auto	Cyst	Met	Hem	Sensitivity
Cyst	52	1	0	98.1%
Met	2	44	18	68.7%
Hem	0	18	47	72.3%
Specificity	98.4%	83.9%	84.6%	

With GAN

True \ Auto	Cyst	Met	Hem	Sensitivity
Cyst	53	0	0	100%
Met	2	52	10	81.2%
Hem	1	13	51	78.5%
Specificity	97.7%	89%	91.4%	

Training GAN with only 182 samples?

Train each lesion class separately (for GAN)

Using the same 3-fold cross validation process and the same data partition.

Use classic augmentation for the training process

Challenge human experts

They challenged two radiologists

- to classify real and fake lesion ROIs into one of three classes: cyst, metastasis or hemangioma.

The goal of the experiment was

- to check if the radiologists would perform differently on a real lesion vs. a fake one

Results:

- Expert 1: 78% vs 77.5%
- Expert 2: 69.2% vs 69.2%

Summary

- Can use classic augmentation to generate training data for GAN.
 - But sometimes this does not work
- Combine classic augmentation with GAN synthesised samples may improve accuracy in some situations.

GAN augmentation: Augmenting training data using generative adversarial networks

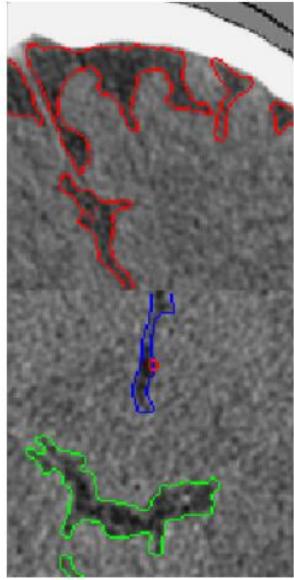
Bowles, Christopher, et al. "Gan augmentation: Augmenting training data using generative adversarial networks." arXiv preprint arXiv:1810.10863 (2018).

Bowles et al. 使用 GAN 增強在 CT 上的 CSF segmentation 問題

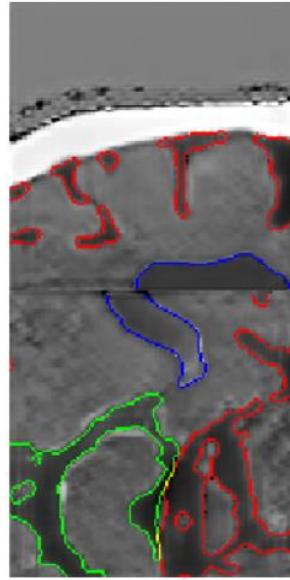
- 使用 Progressive GAN 來生成資料
- 搭配 rotation augmentation

看看在不同數量的資料下

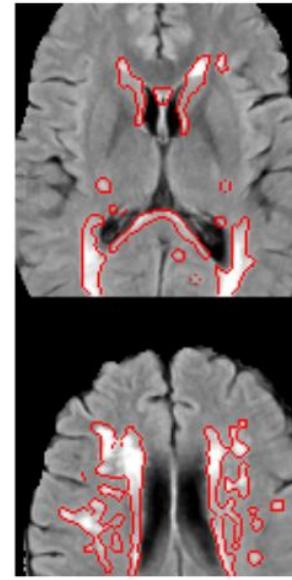
- Dice Similarity Coefficient 的表現。
- 大致上 GAN 會幫忙增加 1~2%



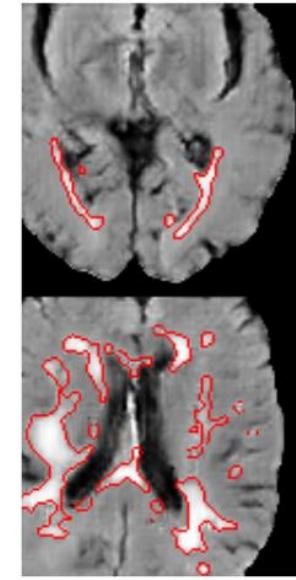
(a) Real CT



(b) Synthetic CT



(c) Real MRI



(d) Synthetic MRI

Fig. 1: Examples of real and GAN generated synthetic patches. *Left*: CSF. Red: Cortical CSF. Green: Brain stem CSF. Blue: Ventricular CSF. *Right*: WMH.

<u>Aa</u> Available data	# 100%	# 50%	# 10%
No augmentation	88.1	85	75.1
GAN augmentation	88.4	85.6	76.3
Rotation augmentation	88.9	86	76.9
GAN + Rotation augmentation	89.3	86.9	78.4

Table 1: Summary of experiments

% of available real data sampled from	% added synthetic data	Segmentation network	Dataset	Augmentation type
100, 50, 10	0, 50, 100	UNet, UResNet	CT	Rotation+GAN
100, 50, 10	0, 100	UNet	CT	None, GAN, rotation, rotation+GAN
100, 50, 10	0, 12.5, 25, 37.5, 50, 100	UNet	CT	Rotation+GAN
100, 90...20, 10	0, 50	UNet	CT	Rotation+GAN
100, 50, 10	0, 50, 100	DeepMedic	MR	GAN

Table 2: **CSF segmentation on CT**: Results with different proportions of the available training data and varying amounts of additional synthetic data using UNet and UResNet architectures.

		Available data					
		UNet			UResNet		
		100%	50%	10%	100%	50%	10%
Additional Data	0%	88.9 (0.51)	86.0 (0.50)	76.9 (0.58)	86.8 (0.82)	82.7 (1.55)	72.5 (1.98)
	50%	89.2 (0.30)	87.3 (0.46)	78.6 (1.04)	86.3 (1.44)	84.3 (1.31)	74.3 (1.63)
	100%	89.3 (0.39)	86.9 (0.36)	78.4 (0.99)	86.3 (1.24)	84.1 (1.32)	74.7 (1.18)

Table 3: **CSF segmentation on CT**: UNet results with different proportions of the available training data and different augmentation techniques.

	Available data		
	100%	50%	10%
No augmentation	88.1 (0.32)	85.0 (0.58)	75.1 (0.60)
GAN augmentation	88.4 (0.41)	85.6 (1.33)	76.3 (1.77)
Rotation augmentation	88.9 (0.51)	86.0 (0.50)	76.9 (0.58)
GAN + Rotation augmentation	89.3 (0.39)	86.9 (0.36)	78.4 (0.99)

Summary

- This is what you may expected when using the naive GAN augmentation method.
- GAN + classic augmentation usually improves the result a little bit
 - Some times, this is exactly what you want, the last inch of the last mile.

Skin Lesion Classification Using GAN based Data Augmentation

Rashid, H., Tanveer, M. A., & Khan, H. A. (2019, July). Skin lesion classification using GAN based data augmentation. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 916-919). IEEE.

Rashid et al. 使用 skin lesion ISIC 2018 challenge 資料

- 有 7 個類別，7500 筆訓練資料，
- 類別不平衡，最多的一類 NV 有 5342 筆資料，最少的 DF 有 90 筆。

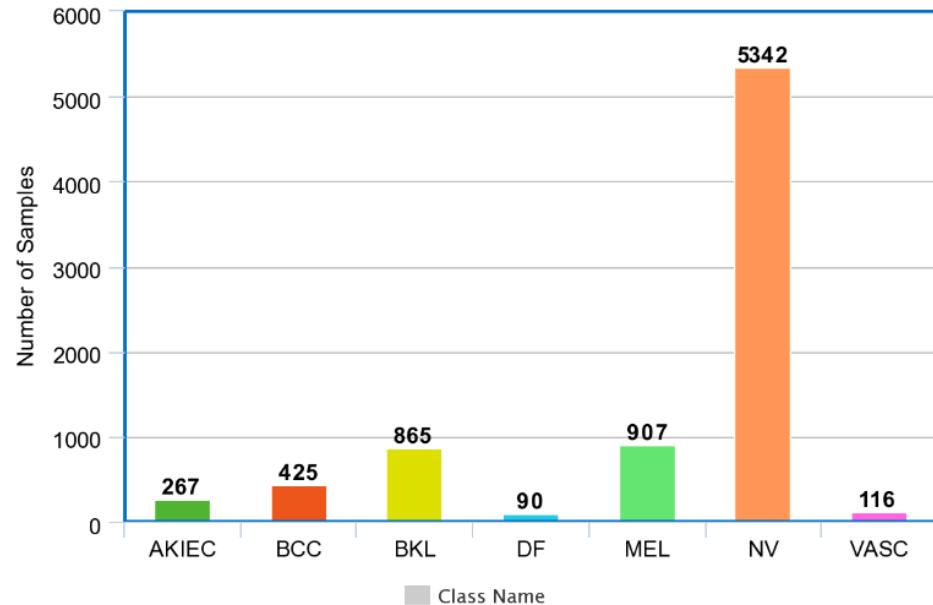
雖然標題是 Data augmentation

- 但實際上使用的是 semi-supervised GAN 架構。
- 也就是 discriminator 同時用原始資料去訓練 classification。

得到的結果是準確度優於單純訓練 DenseNet 及 ResNet-50 classifier

Dataset: Skin lesion ISIC 2018 challenge

- 10k samples, 7 classes
 - Melanoma (MEL);
 - Melanocytic Nevus (NV);
 - Basal Cell Carcinoma (BCC);
 - Actinic Keratosis (AKIEC);
 - Benign Keratosis (BKL);
 - Dermatofibroma (DF);
 - Vascular Lesion (VASC).
- Training 8K/Testing 2K

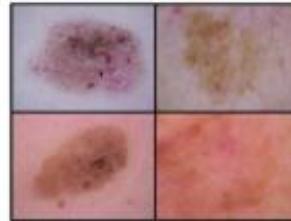




Actinic Keratosis
(AKIEC)



Basal Cell Carcinoma
(BCC)



Benign Keratosis
(BKL)



Dermatofibroma
(DF)



Melanoma
(MEL)



Melanocytic Nevus
(NV)



Vascular Lesion
(VASC)

ACGAN

- <https://arxiv.org/abs/1610.09585>

InfoGAN

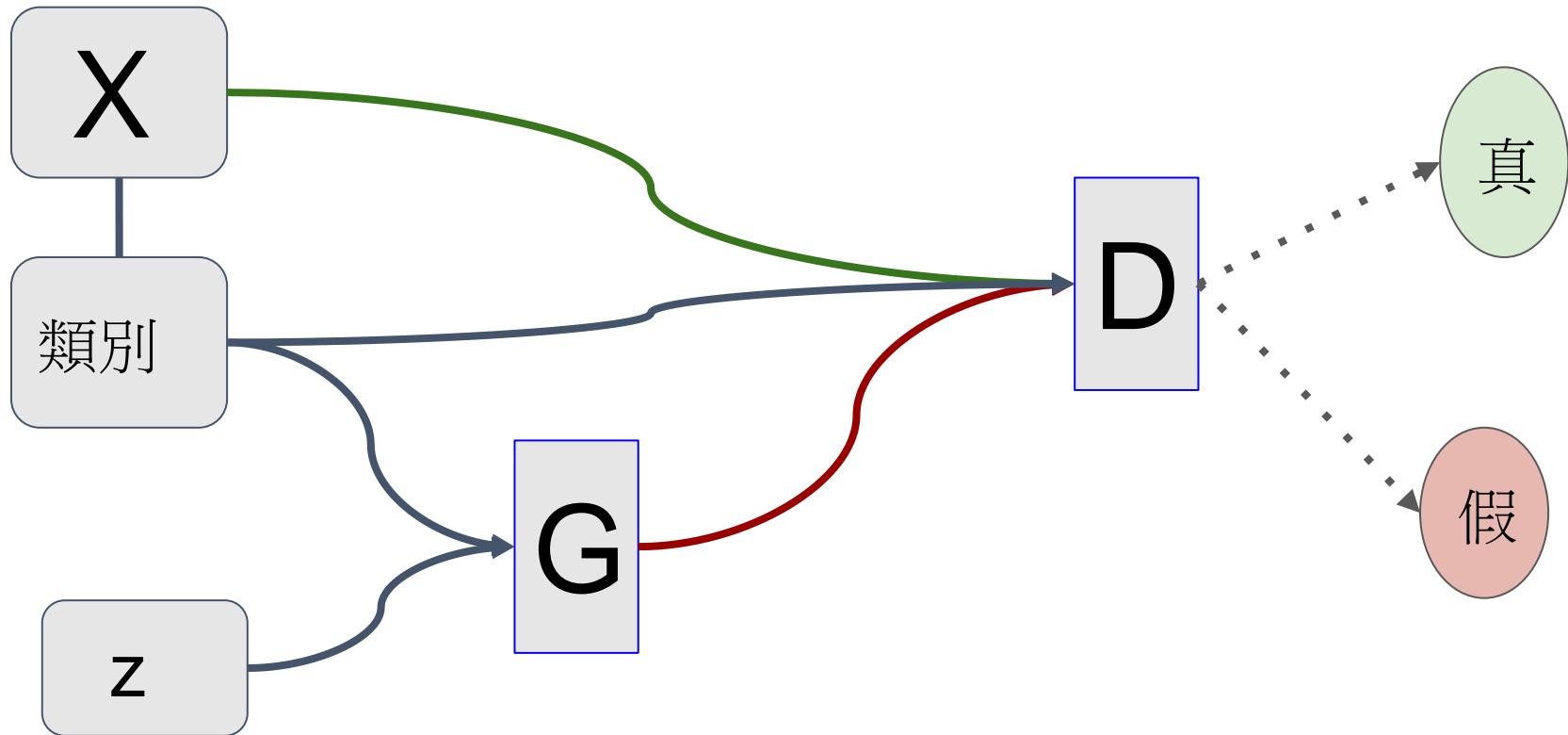
- <https://arxiv.org/abs/1606.03657>

Conditional GAN

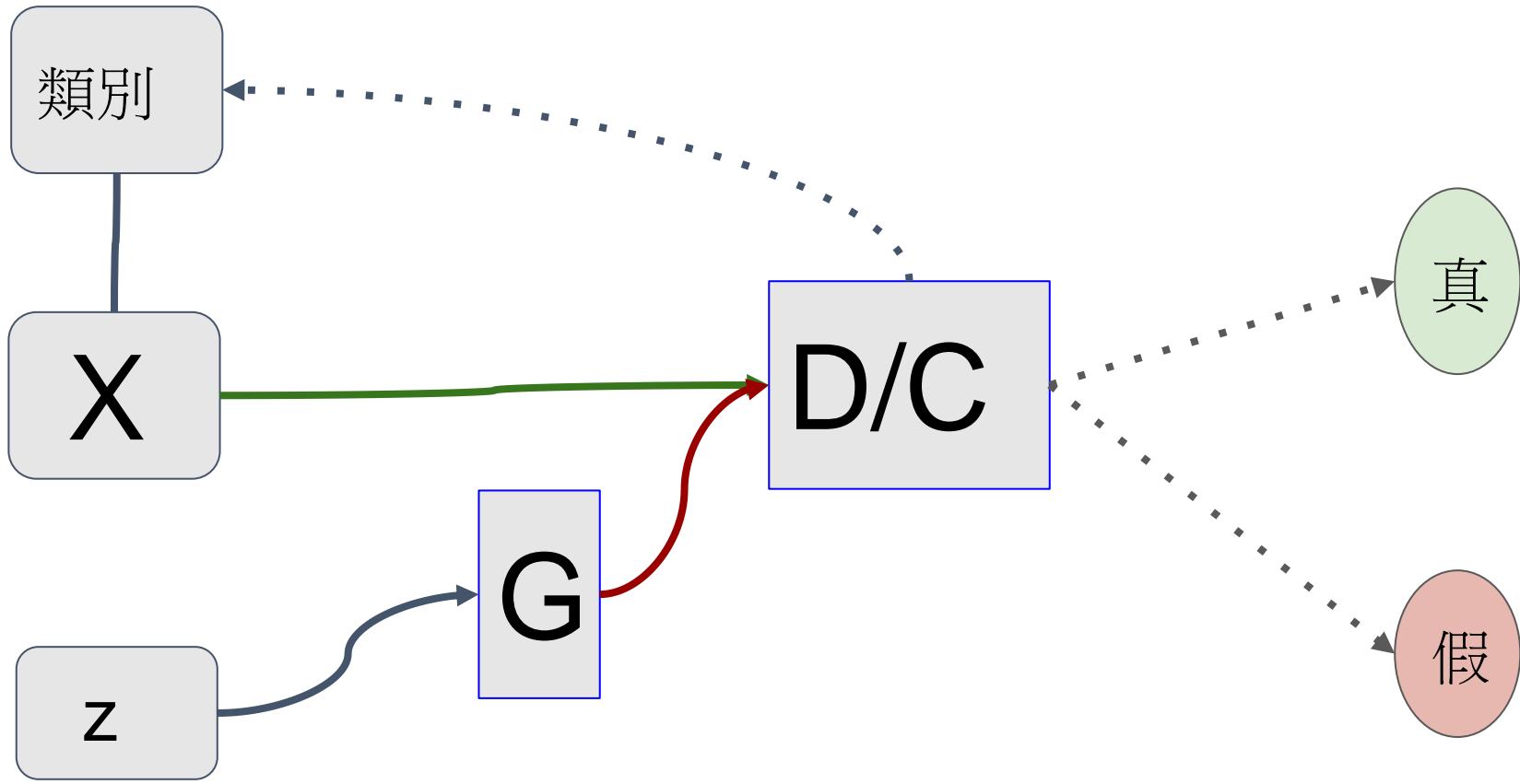
- <https://arxiv.org/abs/1411.1784>

Semi-supervised GAN

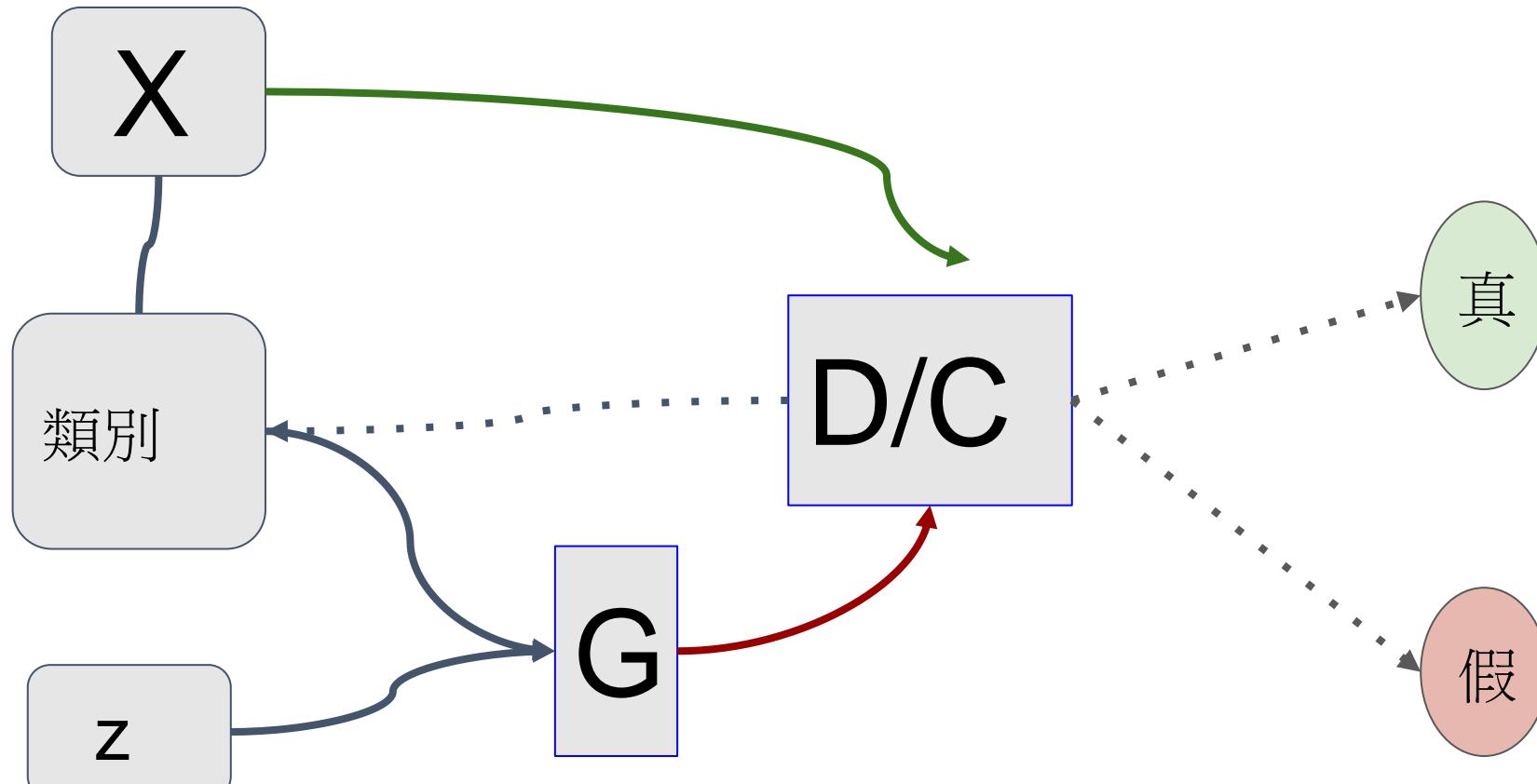
- <https://arxiv.org/abs/1606.01583>

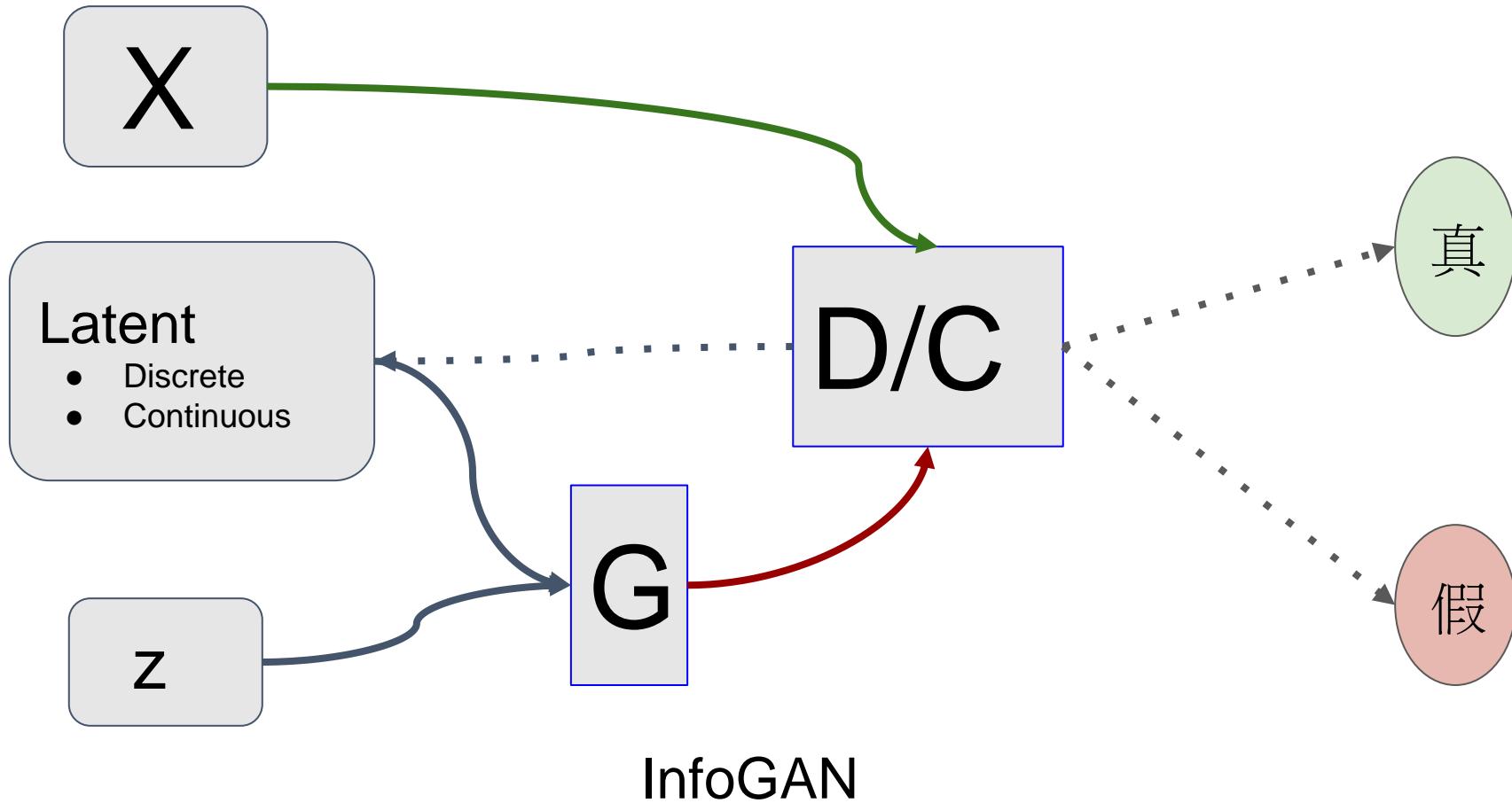


Conditional GAN

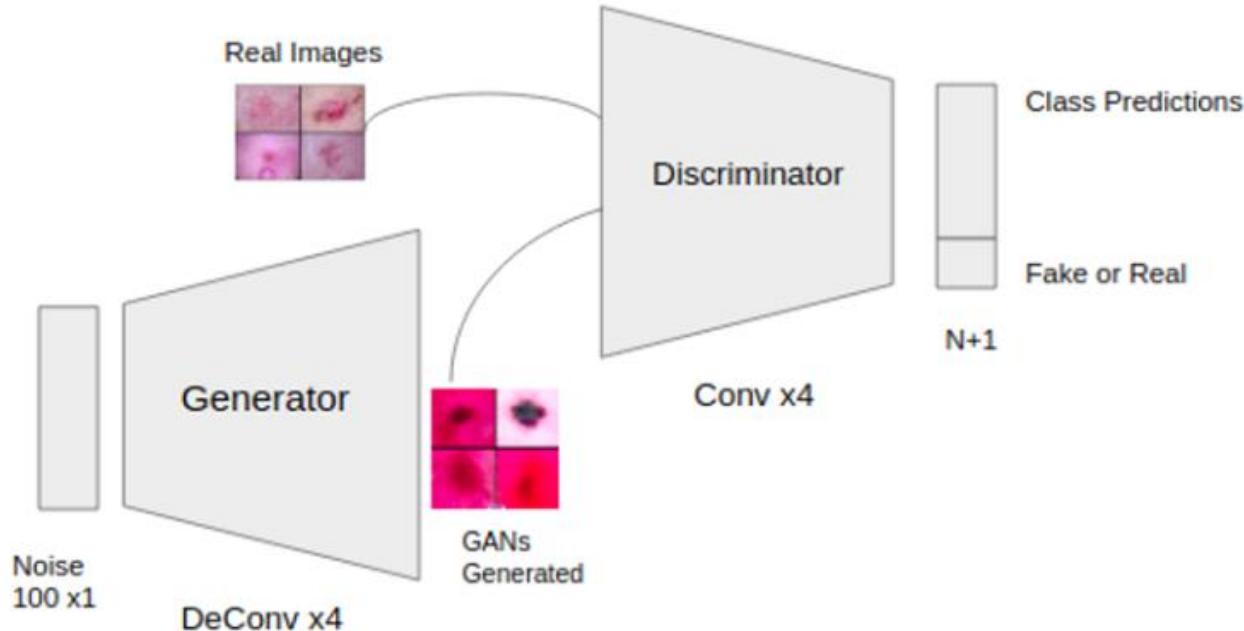


Semi-Supervised GAN





Architecture: Semi-supervised GAN



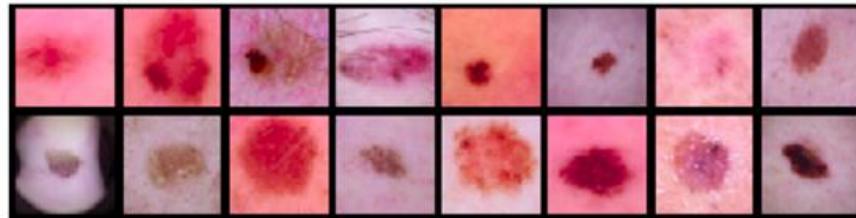
Experiment Setting

Image size 224x224

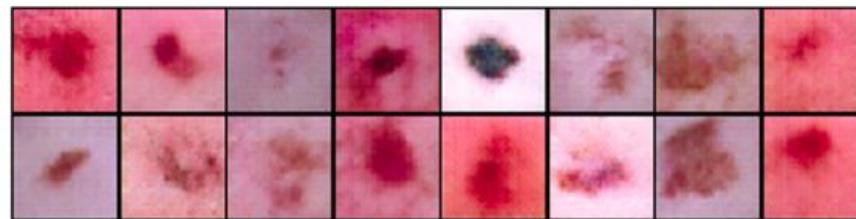
Conventional data augmentation strategies:

- random cropping
- Gaussian blurring
- addition of salt & pepper noise

Synthetic images



(a) Real images sampled from dataset



(b) GAN generated, synthetic images

Fig. 4. Real and GAN generated synthetic images.

Result

Class	Precision	Recall	F1-score
ACKIEC	0.73	0.69	0.710
BCC	0.85	0.90	0.870
BKL	0.80	0.81	0.804
DF	0.89	0.74	0.808
MEL	0.81	0.79	0.800
NV	0.94	0.95	0.945
VASC	0.89	0.92	0.904

TABLE II
PERFORMANCE COMPARISON OF VARIOUS ALGORITHMS ON ISIC 2018
DATASET

Approach	Balance Accuracy Score
GANs	0.861
DenseNet	0.815
ResNet-50	0.792

Summary

- There are other ways of using GAN to improve the result of other tasks.
- GAN is a way of doing self-supervised learning

Cycle GAN-Based Data Augmentation For Multi-Organ Detection In CT Images Via Yolo

Hammami, Maryam, Denis Friboulet, and Razmig Kechichian. "Cycle GAN-Based Data Augmentation For Multi-Organ Detection In CT Images Via Yolo." 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020.

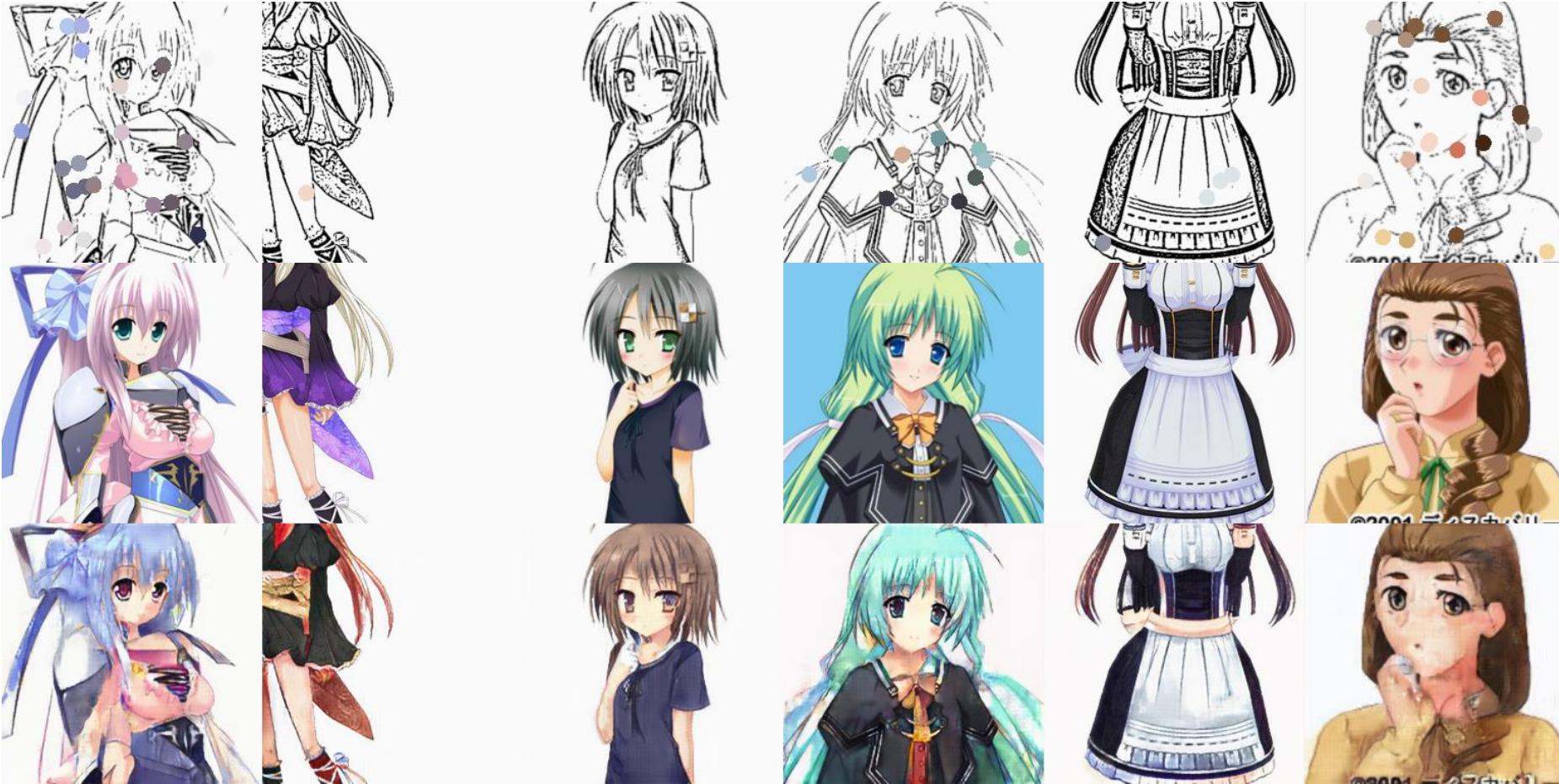
Hammami et al. 處理的問題是 3D CT 上的 Object detection(YOLO).

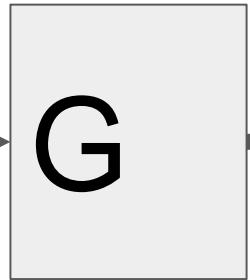
主要是利用 Cycle GAN 把 MRI 轉成 CT 來增加資料數量。

用 mean distance per organ 來來比較

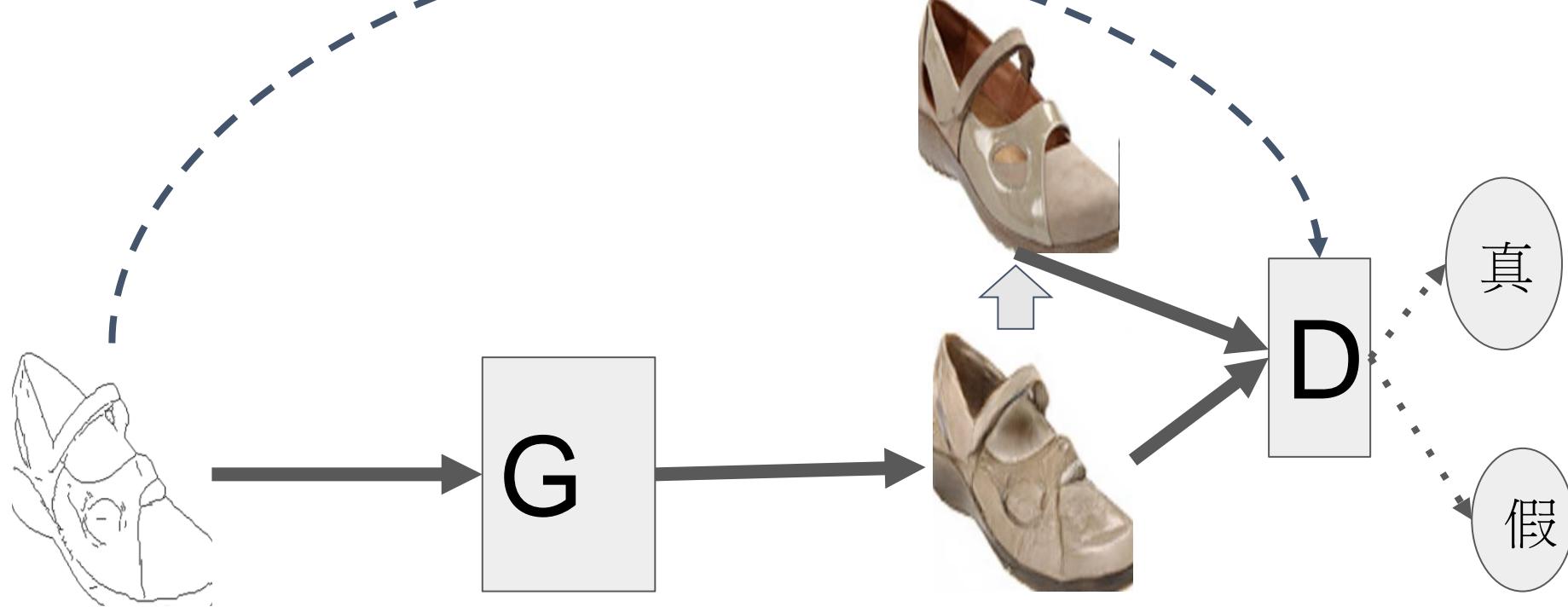
YOLO vs CycleGAN+YOLO

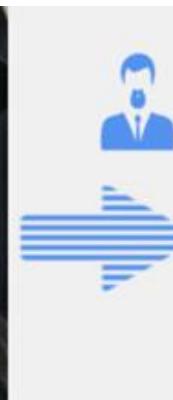
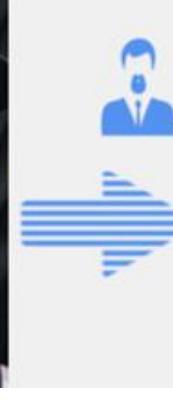
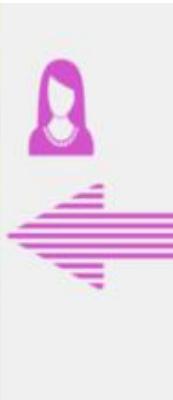
Aa Method	⠇ Liver	⠇ Kidney R	⠇ Kidney L	⠇ Spleen	⠇ Gallbladder
YOLO	7.4 ± 4.4	5.6 ± 12.9	4.7 ± 5.8	6.8 ± 7.0	6.9 ± 10.9
CycleGAN + Yolo	6.9 ± 3.4	5.9 ± 12.4	4.3 ± 4.8	6.5 ± 6.2	7.4 ± 11.2





$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$





Paired

x_i y_i



⋮

Unpaired

X



⋮

Y



⋮

CycleGAN

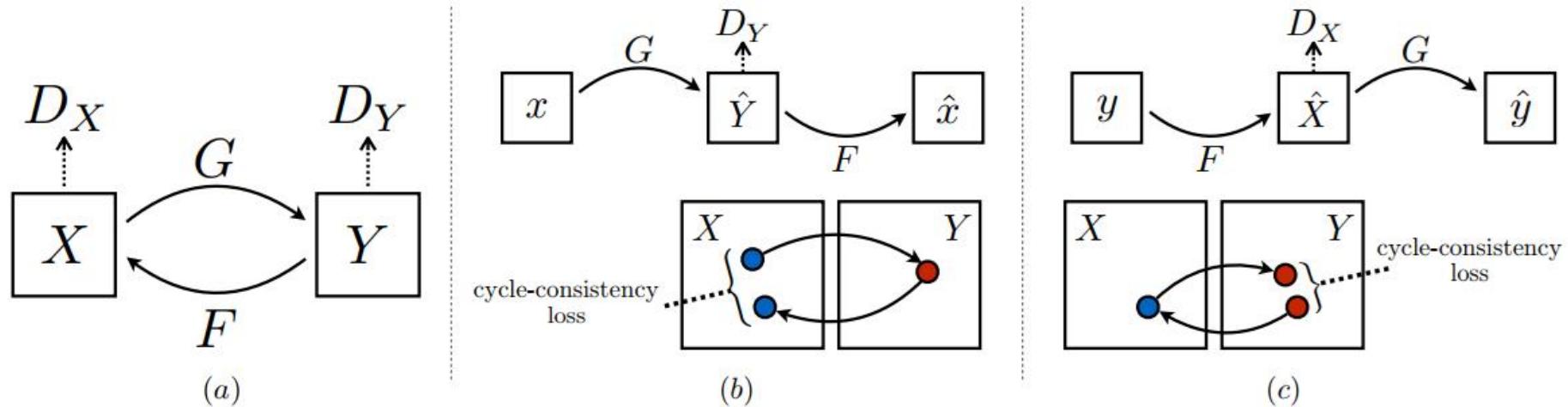


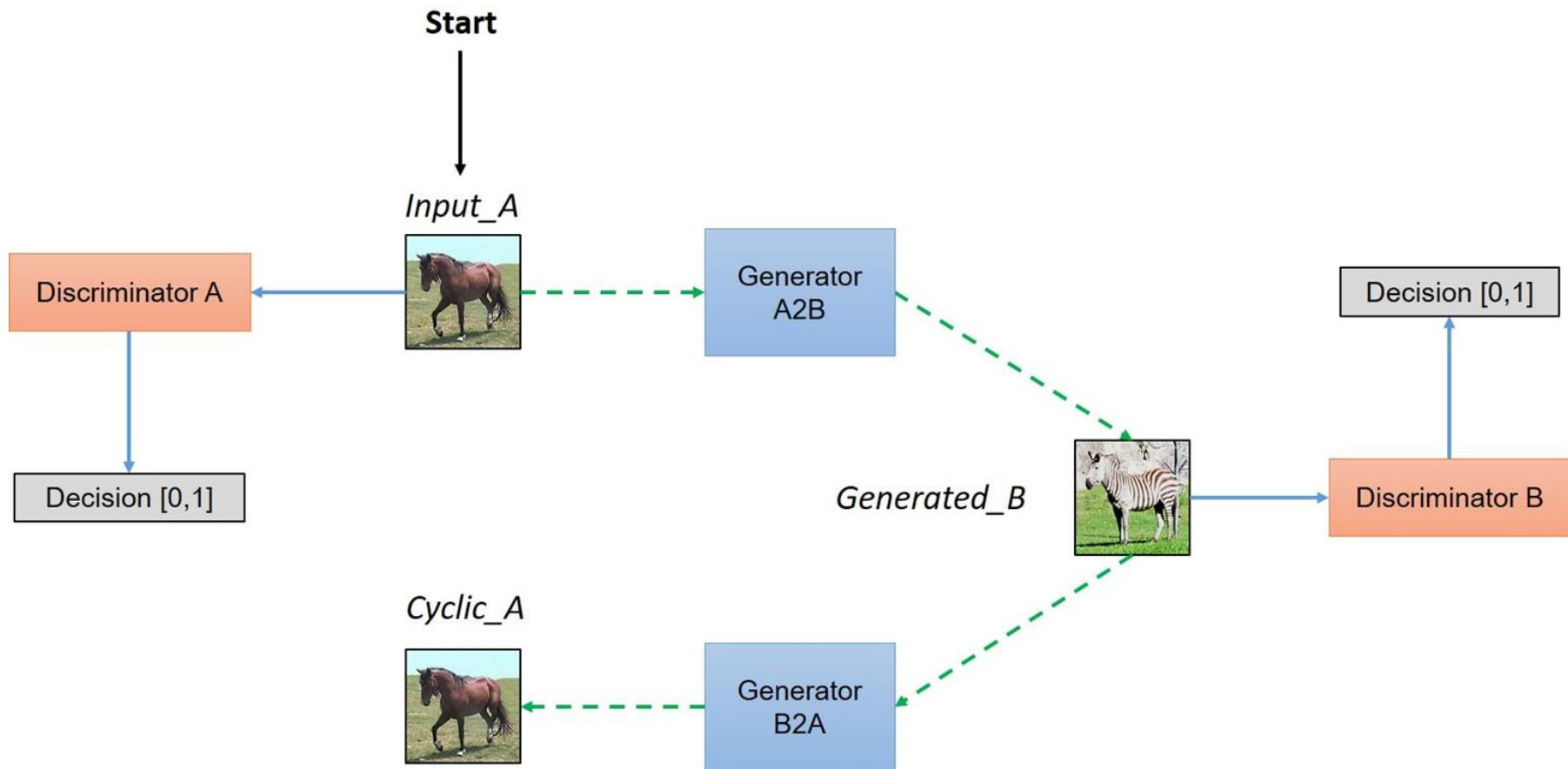
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

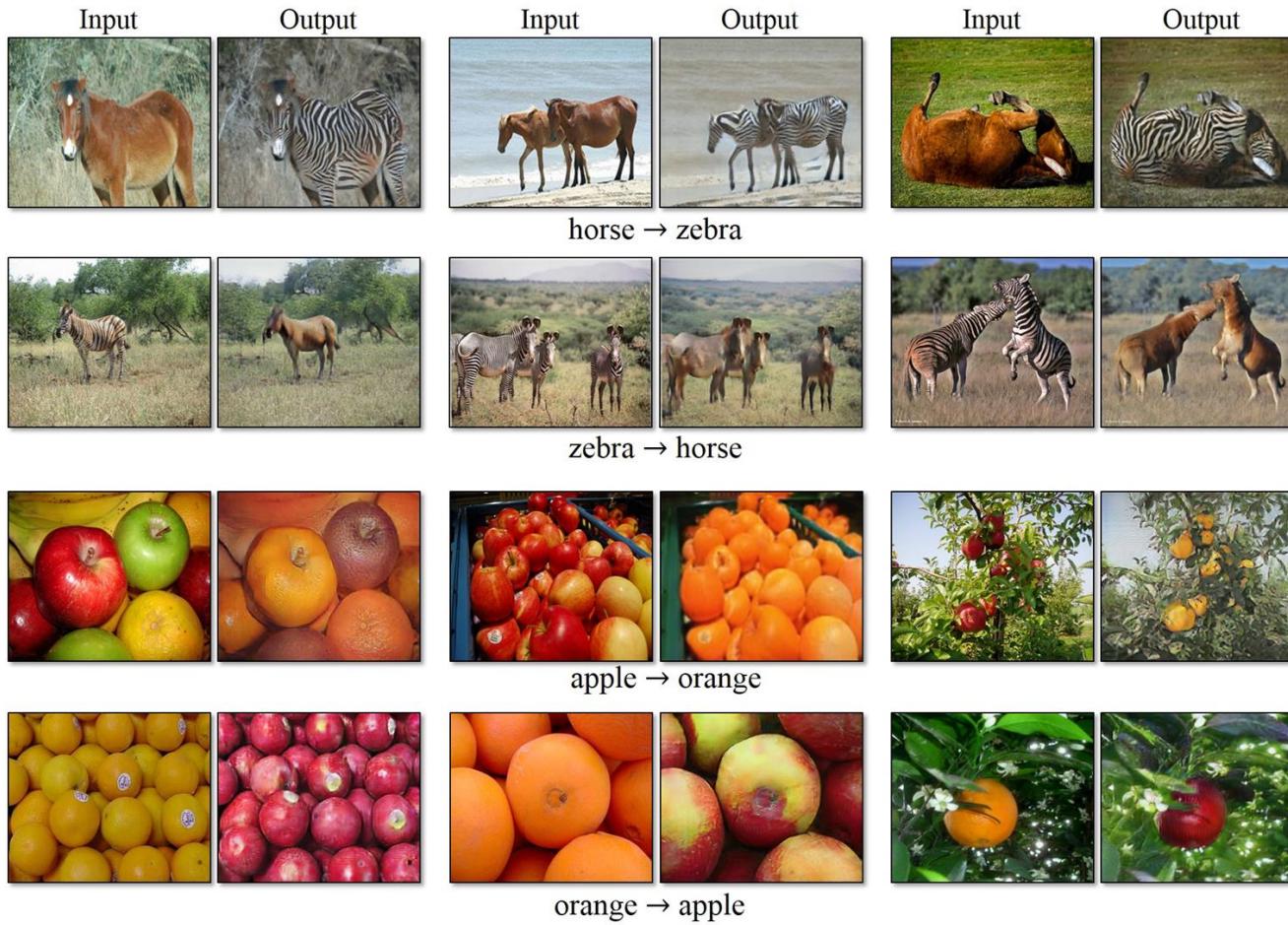
Jun-Yan Zhu*, Taesung Park*, Phillip Isola, Alexei A. Efros Berkeley AI Research Lab, UC Berkeley

$$G(x) \approx Y, \quad F(y) \approx X$$

$$F(G(x)) = x, \quad G(F(y)) = y$$







Dataset

Gold dataset

- the annotations of which were created using manual segmentation

Silver dataset

- the labels of which were obtained by merging the segmentations produced by the algorithms of benchmark participants.

Both datasets consist of

- unpaired 3D contrast-enhanced thoracic-abdominal CT and
- abdominal MRI images providing 20 and 15 structure annotations respectively.

Mean image dimensions are

- 512x512x438 voxels for CT and
- 312x72x384 voxels for MRI.

Consists

- The Gold dataset provides 20 patients per modality.
- Choose 30 patients per modality in the Silver dataset.

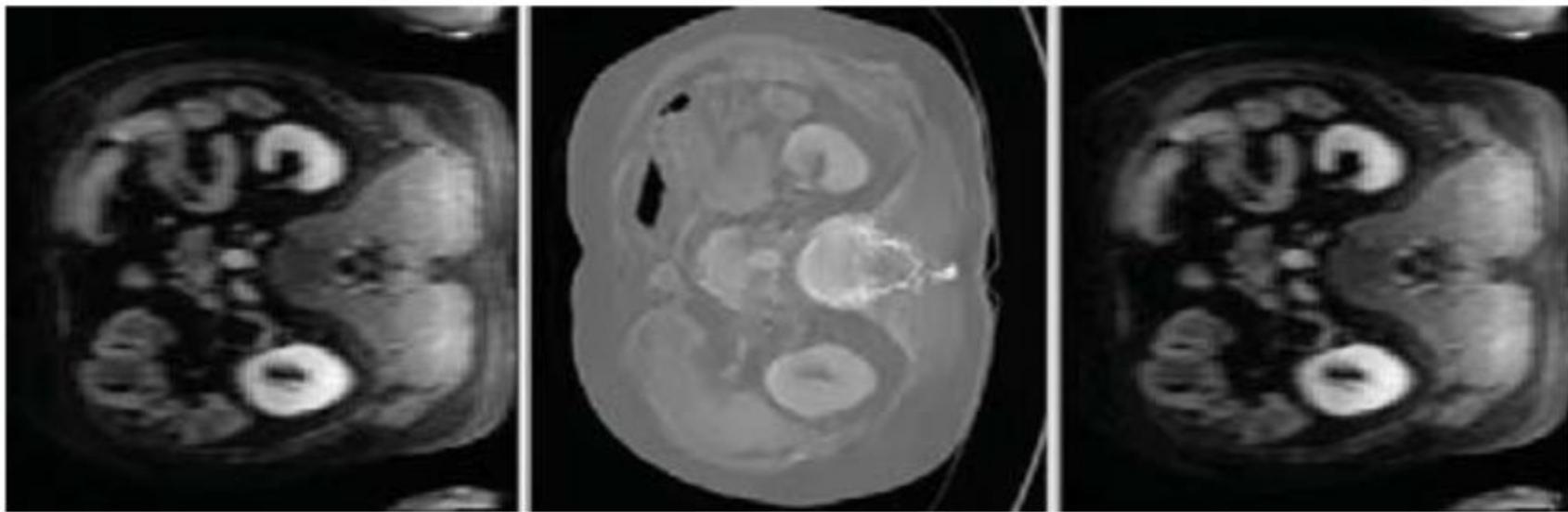
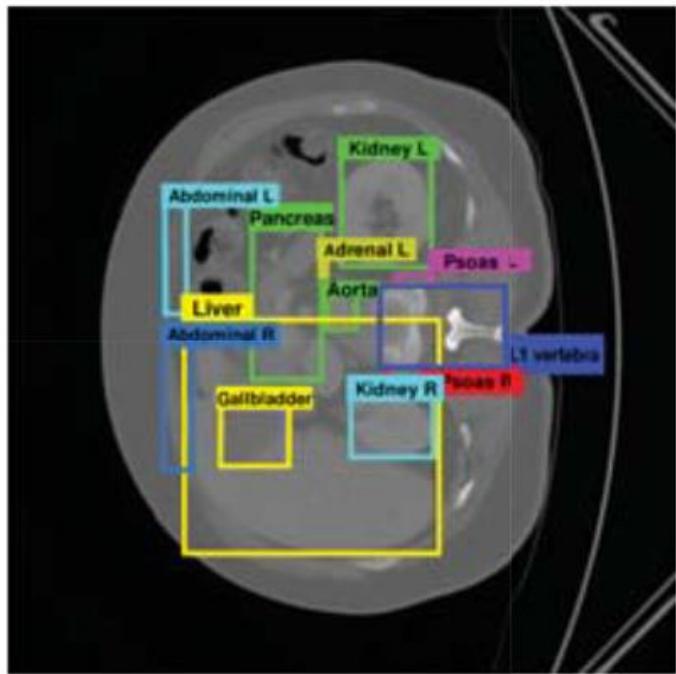
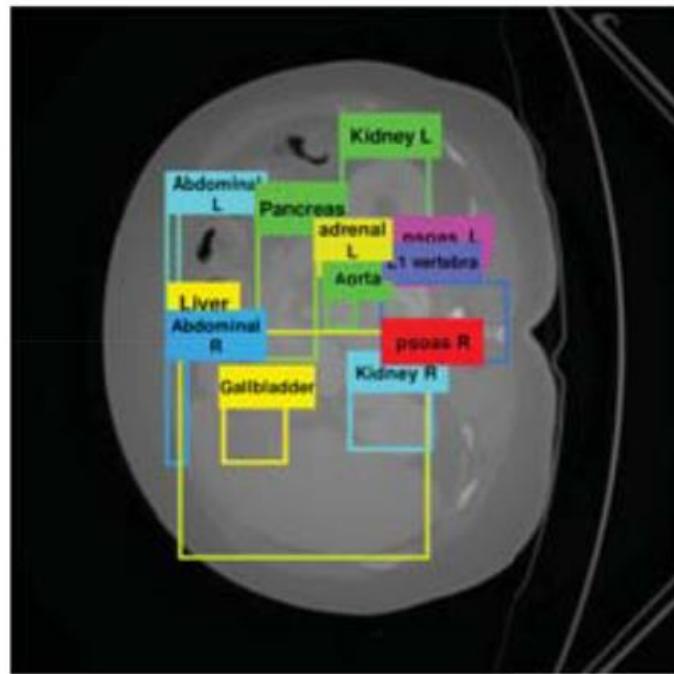


Fig. 2. Qualitative results of cross-modality generation (from MRI to CT image). The real MRI image (left), the generated CT image (center) and the reconstructed MRI image (right).



(a) ground-truth



(b) prediction

Fig. 3. 2D multi-organ detection on an axial CT image.

Experiment Setting

10 fold cross validation

- under two scenarios, with or without data augmentation.

a CycleGAN trained on the Silver dataset

- was used to translate MRI images in the Gold dataset into CT
- which were used to augment the training data in each of the 10 folds.

Table 2. Comparison with state-of-the-art methods based on mean distances per organ.

Method	Liver	Kidney R	Kidney L	Spleen	Gallbladder
Cuingnet [1]	12.2 ± 4	6.4 ± 4	6.8 ± 6	9.0 ± 5	11.8 ± 8
Criminsi [2]	14.0 ± 5	13.2 ± 6	12.3 ± 7	14.2 ± 6	15.5 ± 8
Gauriau [3]	10.7 ± 4	5.6 ± 3	5.5 ± 4	7.9 ± 4	9.5 ± 4
YOLO	7.4 ± 4.4	5.6 ± 12.9	4.7 ± 5.8	6.8 ± 7.0	6.9 ± 10.9
CycleGAN + Y.	6.9 ± 3.4	5.9 ± 12.4	4.3 ± 4.8	6.5 ± 6.2	7.4 ± 11.2

Table 1. YOLO (mean dist. 8.66 mm) vs CycleGAN+YOLO (mean dist. 7.95 mm) comparison on per organ mean distance.

	Pancreas	Gallbladder	Bladder	Verteba L1	Kidney R
YOLO	14.3 ± 10.4	6.9 ± 10.9	4.0 ± 1.2	6.2 ± 3.5	5.6 ± 12.9
CycleGAN + Y.	10.6 ± 5.2	7.4 ± 11.2	4.5 ± 1.6	5.8 ± 3.3	5.9 ± 12.4
	Kidney L	Adrenal R	Adrenal L	Psoas R	Psoas L
YOLO	4.7 ± 5.8	6.6 ± 6.5	8.1 ± 8.4	16.6 ± 13.4	12.7 ± 7.1
CycleGAN + Y.	4.3 ± 4.8	6.3 ± 5.9	7.8 ± 8.7	11.8 ± 6.9	12.8 ± 5.7
	Abdominal R	Abdominal L	Aorta	Liver	Spleen
YOLO	13.6 ± 12.1	11.9 ± 7.3	4.0 ± 3.0	7.4 ± 4.4	6.8 ± 7.0
CycleGAN + Y.	11.9 ± 6.7	12.2 ± 7.7	3.9 ± 2.6	6.9 ± 3.4	6.5 ± 6.2

Summary

- Image to image translation requires fewer training data, compares to unconditional GAN.
- There are more ways to use GAN.

AnoGAN

- Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., Langs, G., 2017a. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Niethammer, M., Styner, M., Aylward, S., Zhu, H., Oguz, I., Yap, P.-T., Shen, D. (Eds.), Information Processing in Medical Imaging. Springer International Publishing, Cham, pp. 146–157.
- Schlegl, Thomas, et al. "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks." Medical image analysis 54 (2019): 30-44.
- Shin, Dong-Hoon, Roy C. Park, and Kyungyong Chung. "Decision Boundary-Based Anomaly Detection Model using Improved AnoGAN from ECG Data." IEEE Access 8 (2020): 108664-108674.
- Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chan-drasekhar, V. R. **Efficient GAN-Based Anomaly Detection**. abs/1802.06222, 2018. <http://arxiv.org/abs/1802.06222>.
- Di Mattia, Federico, et al. "A survey on gans for anomaly detection." arXiv preprint arXiv:1906.11632 (2019).
- Akcay, Samet, Amir Atapour-Abarghouei, and Toby P. Breckon. "**Ganomaly**: Semi-supervised anomaly detection via adversarial training." Asian conference on computer vision. Springer, Cham, 2018.



StyleGAN Encoder



AnoGAN

利用健康醫療影像為樣本訓練 GAN, 用來偵測異常。

- 紿定圖片 x 和 latent vector z ,
- 定義 Residual Loss $R(x, z) = |x - G(z)|$ (where G is the generator),
- Discriminator loss $D(x, z) = |f(x) - f(G(z))|$, where f is features from the discriminator.
- 固定 x , 極小化 $A(x, z) = (1 - \lambda)R(x, z) + \lambda D(x, z)$ (for some fixed λ)
- 我們可以找到對應到 x 的 latent z .
- 而 $\min_z A(x, z)$ 可以用來衡量 x 是否在原來的 distribution 中。

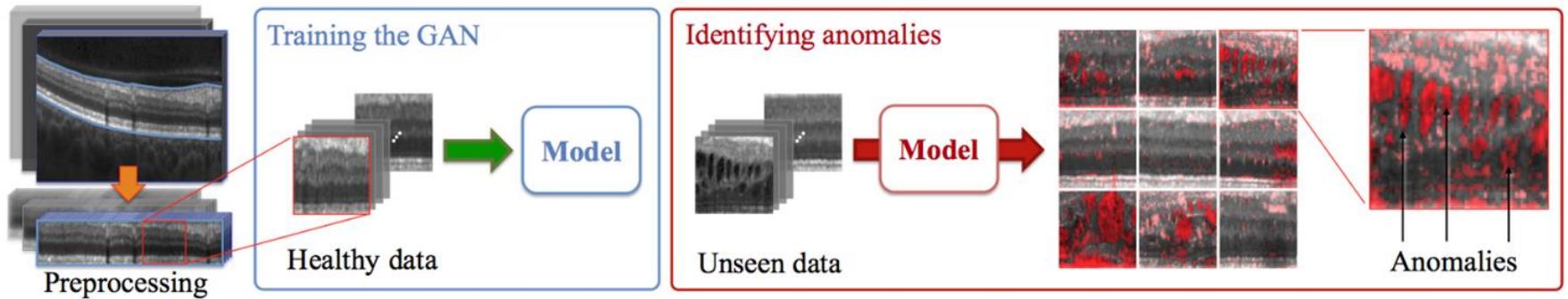
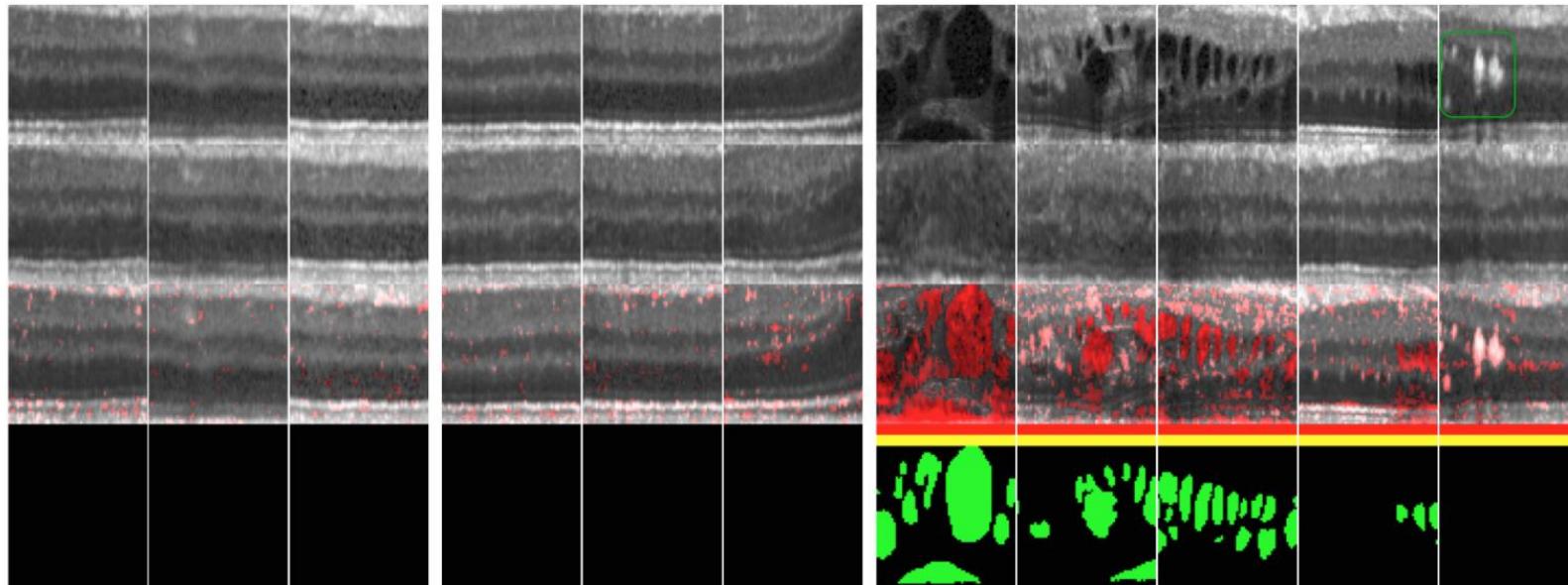


Fig. 1. Anomaly detection framework. The preprocessing step includes extraction and flattening of the retinal area, patch extraction and intensity normalization. Generative adversarial training is performed on healthy data and testing is performed on both, unseen healthy cases and anomalous data.

- Training G, D
- Given x , find a z so that $G(z) \sim x$
 - Minimize $(1 - \lambda) |x - G(z)| + \lambda|f(x) - f(G(z))|$, where f is features from the discriminator.
- If $G(z)$ is not similar enough to x , then x is likely not on the manifold, hence an anomaly.
- $|x - G(z)|$ is used for the identification of anomalous regions.

Result



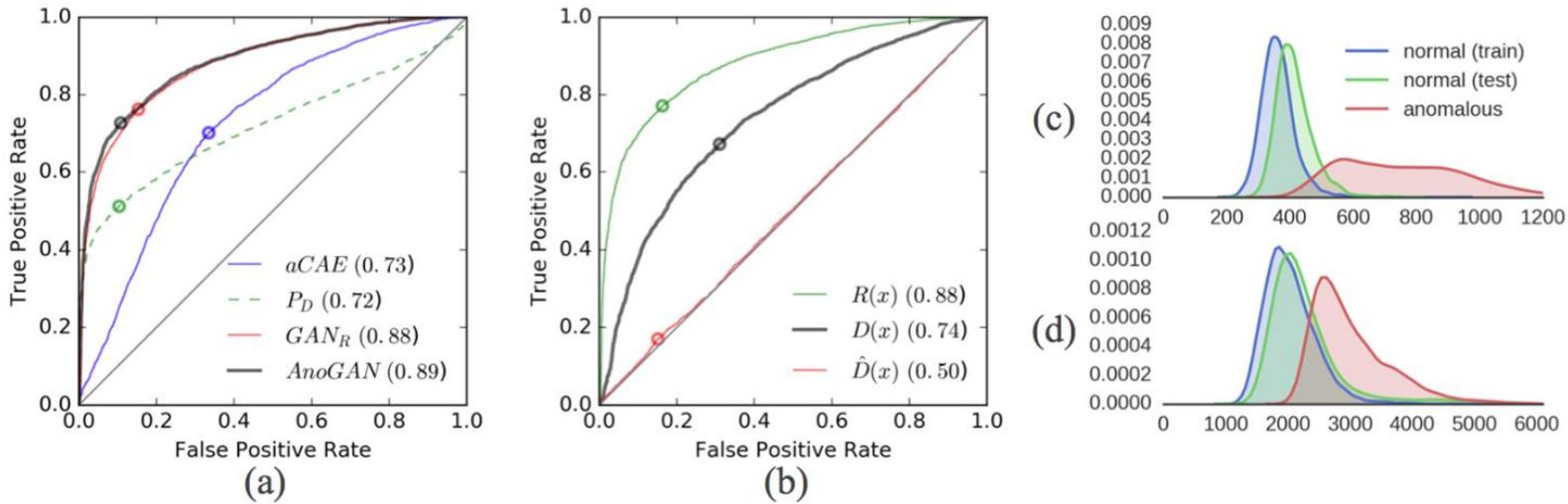


Fig. 4. Image level anomaly detection performance and suitability evaluation. (a) Model comparison: ROC curves based on $aCAE$ (blue), GAN_R (red), the proposed $AnoGAN$ (black), or on the output P_D of the trained discriminator (green). (b) Anomaly score components: ROC curves based on the *residual score* $R(\mathbf{x})$ (green), the *discrimination score* $D(\mathbf{x})$ (black), or the *reference discrimination score* $\hat{D}(\mathbf{x})$ (red). (c) Distribution of the *residual score* and (d) of the *discrimination score*, evaluated on normal images of the training set (blue) or test set (green), and on images extracted from diseased cases (red).

	Precision	Recall	Sensitivity	Specificity	AUC
aCAE	0.7005	0.7009	0.7011	0.6659	0.73
P_D	0.8471	0.5119	0.5124	0.8970	0.72
GAN_R	0.8482	0.7631	0.7634	0.8477	0.88
AnoGAN	0.8834	0.7277	0.7279	0.8928	0.89

Summary

- GAN is trained on **normal** images
- Use an inverse of the generator.
- The basic idea is simple, but the metric matters.
- The inverse process might be slow.

f-AnoGAN

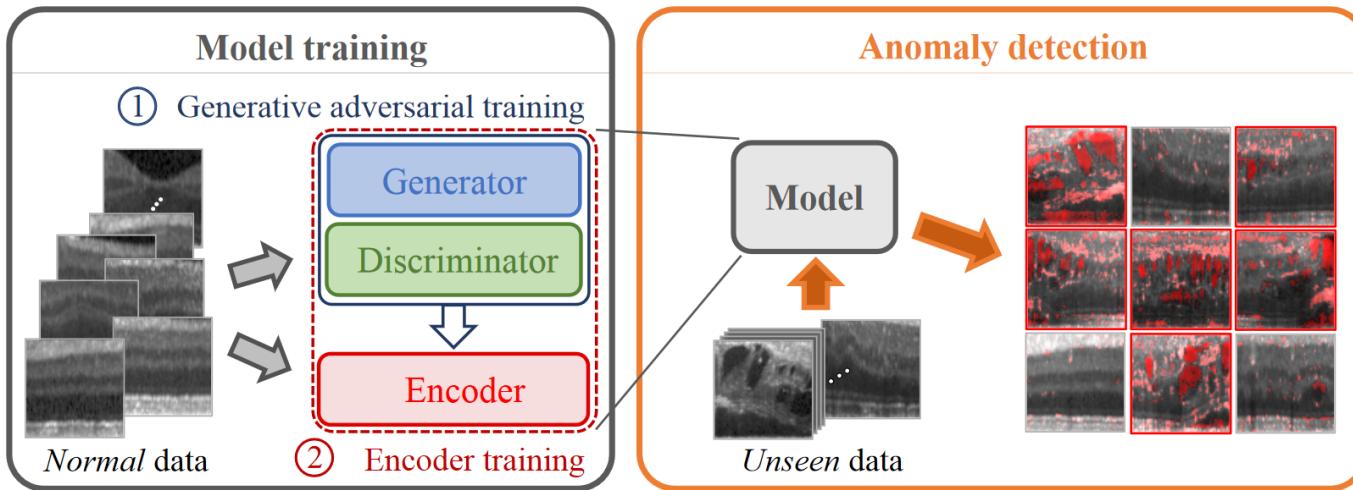


Figure 1: Anomaly detection framework. Both steps of model training, generative adversarial training (yields a trained generator and discriminator) and encoder training (yields a trained encoder), are performed on *normal* (“healthy”) data and anomaly detection is performed on both, unseen healthy cases and anomalous data. (Best viewed in color)

Train WGAN



Train encoders



Inference

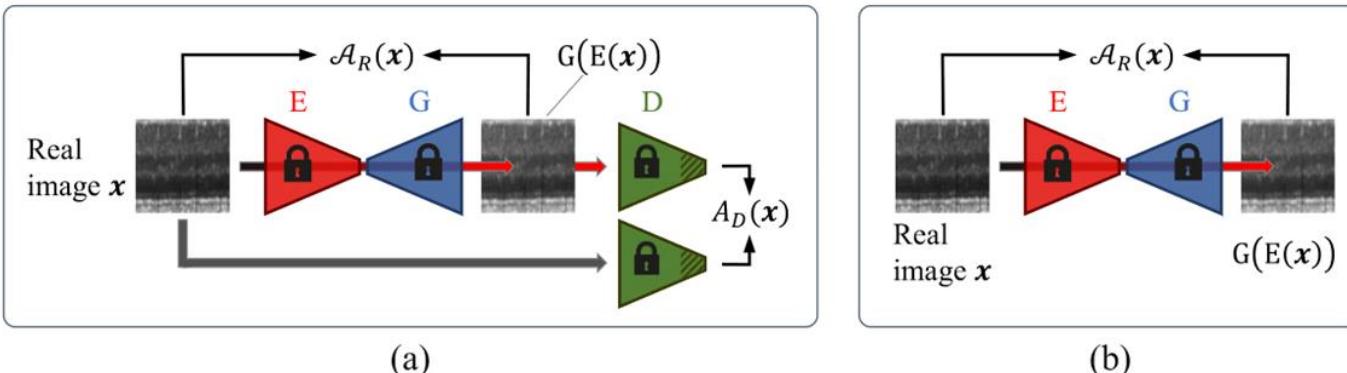


Figure 3: Inference on new images for anomaly scoring. Anomaly scoring uses exactly the same architecture and dataflow used for encoder training. (a) Proposed *f-AnoGAN* model that uses discriminator guided encoder training (*izi_f* architecture). (b) Anomaly quantification for both underlying encoder training architectures that do not include a discriminator based term (*izi* architecture and *ziz* architecture). (Best viewed in color)

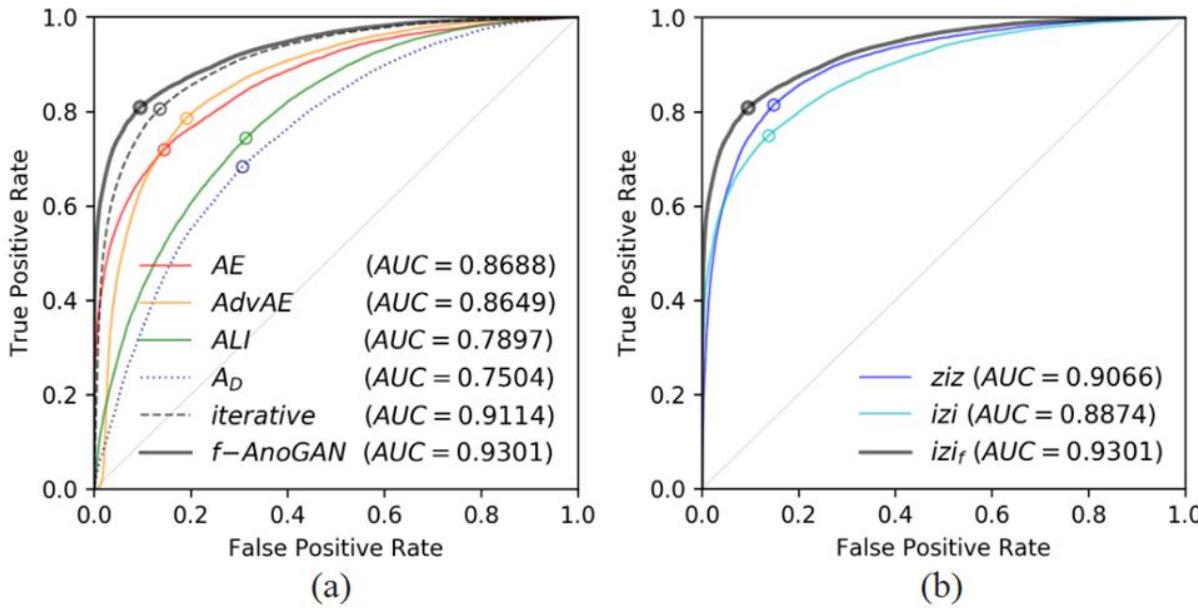


Figure 8: Image-level anomaly detection accuracy evaluation. Comparison of different approaches for image-level anomaly detection based on receiver operating characteristic (ROC) curves and corresponding area under the ROC curve (AUC) values (specified in parentheses). (a) Comparison of the proposed *f-AnoGAN* and alternative approaches: *AE* (red), *AdvAE* (yellow), *ALI* (green), directly utilizing the discriminator's output of the trained WGAN (A_D , blue dotted), *iterative* z-mapping based on a trained WGAN (following *AnoGAN* (Schlegl et al., 2017a), gray dashed), and the proposed *f-AnoGAN* model (gray). (b) Comparison of different encoder training approaches based on the same pre-trained WGAN: *ziz* encoder training (blue), *izi* encoder training (cyan), and *izi_f* encoder training (gray), which is implemented by the proposed *f-AnoGAN* model. (Please find more details on the different approaches in the main text.)

Table 1: Clinical performance statistics calculated at the Youden index of the receiver operating characteristic (ROC) curve, the corresponding area under the ROC curve (AUC) and f-score measuring the image-level anomaly detection performance of a *convolutional autoencoder* (*AE*), *adversarial convolutional autoencoder* (*AdvAE*), *ALI* model, based on the output of the WGAN discriminator (A_D), iterative z-mapping utilizing the trained WGAN model (*iterative*) following Schlegl et al. (2017a), and our proposed *fast AnoGAN* (*f-AnoGAN*).

	Precision	Sensitivity	Specificity	f-score	AUC
AE	0.6824	0.7195	0.8550	0.7005	0.8688
AdvAE	0.6405	0.7856	0.8092	0.7057	0.8649
ALI	0.5063	0.7434	0.6863	0.6023	0.7897
A_D	0.4909	0.6831	0.6931	0.5713	0.7504
iterative	0.7202	0.8049	0.8645	0.7602	0.9114
<i>f-AnoGAN</i>	0.7863	0.8091	0.9049	0.7975	0.9301

Summary

- Use encoder instead of optimization for GAN inversion
- Investigate several naive/straightforward ways of training the encoder

EGBAD

EFFICIENT GAN-BASED ANOMALY DETECTION

- Use BiGAN(aka ALI) to train the encoder

$$V(D, E, G) = \mathbb{E}_{x \sim p_X} [\mathbb{E}_{z \sim p_E(\cdot|x)} [\log D(x, z)]] + \mathbb{E}_{z \sim p_Z} [\mathbb{E}_{x \sim p_G(\cdot|z)} [1 - \log D(x, z)]]$$

BiGAN

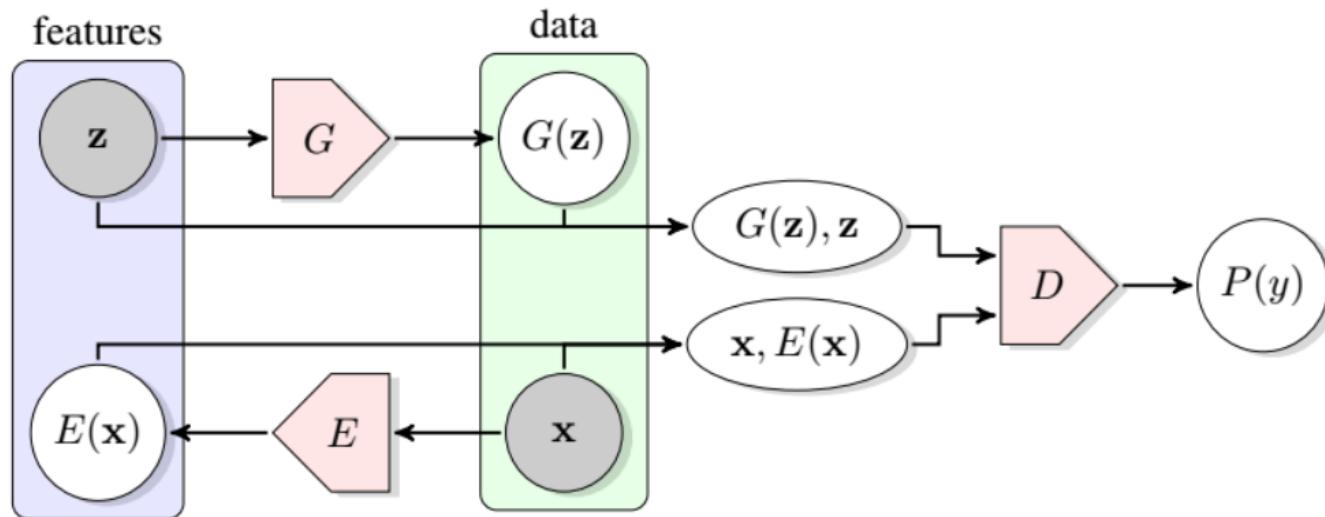
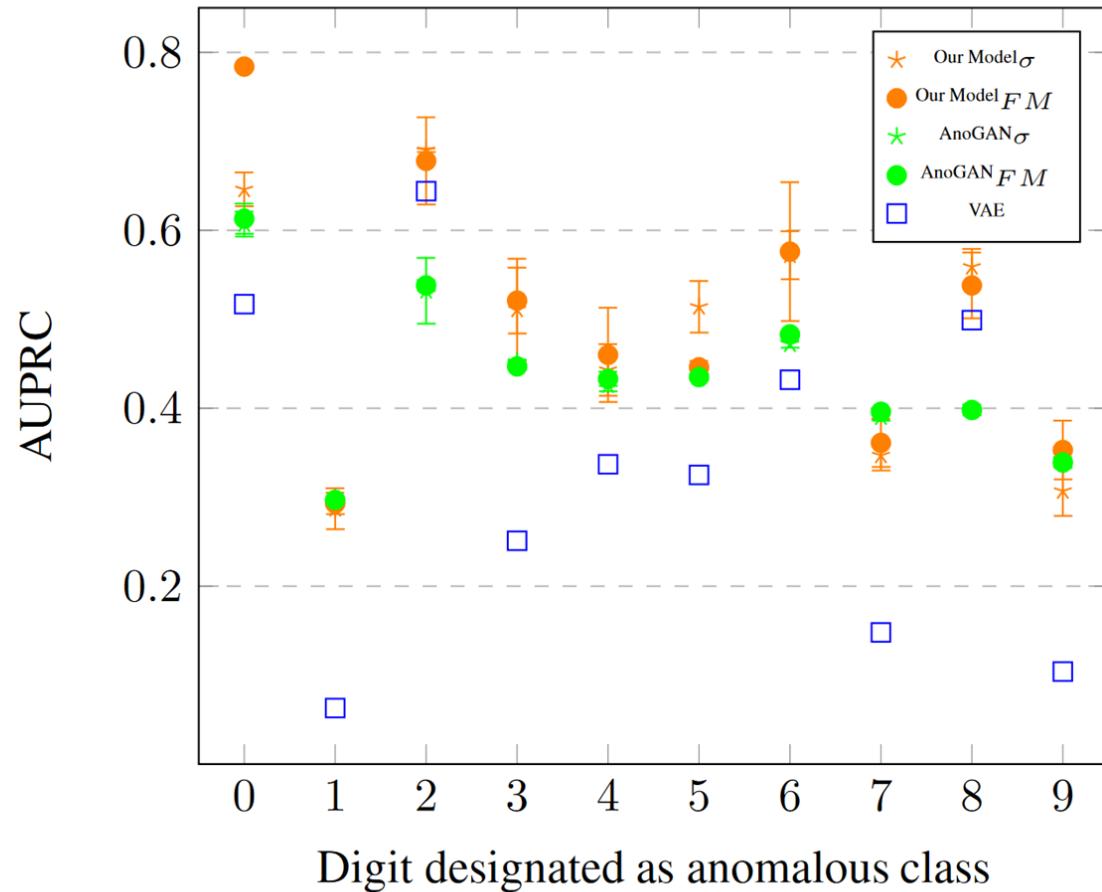


Figure 1. The structure of BiGAN proposed in (Donahue et al., 2016).



Model	Precision	Recall	F1
OC-SVM	0.7457	0.8523	0.7954
DSEBM-r	0.8521	0.6472	0.7328
DSEBM-e	0.8619	0.6446	0.7399
DAGMM-NVI	0.9290	0.9447	0.9368
DAGMM	0.9297	0.9442	0.9369
AnoGAN _{FM}	0.8786 ± 0.0340	0.8297 ± 0.0345	0.8865 ± 0.0343
AnoGAN _{σ}	0.7790 ± 0.1247	0.7914 ± 0.1194	0.7852 ± 0.1181
Our Model _{FM}	0.8698 ± 0.1133	0.9523 ± 0.0224	0.9058 ± 0.0688
Our Model _{σ}	0.9200 ± 0.0740	0.9582 ± 0.0104	0.9372 ± 0.0440

GANomaly

the operating principle of the anomaly detection of this work lies in the **autoencoder** structure

the encoder E at the end of the generator structure helps, during the training phase, to learn to encode the images in order to have the best possible representation of x that could lead to its reconstruction

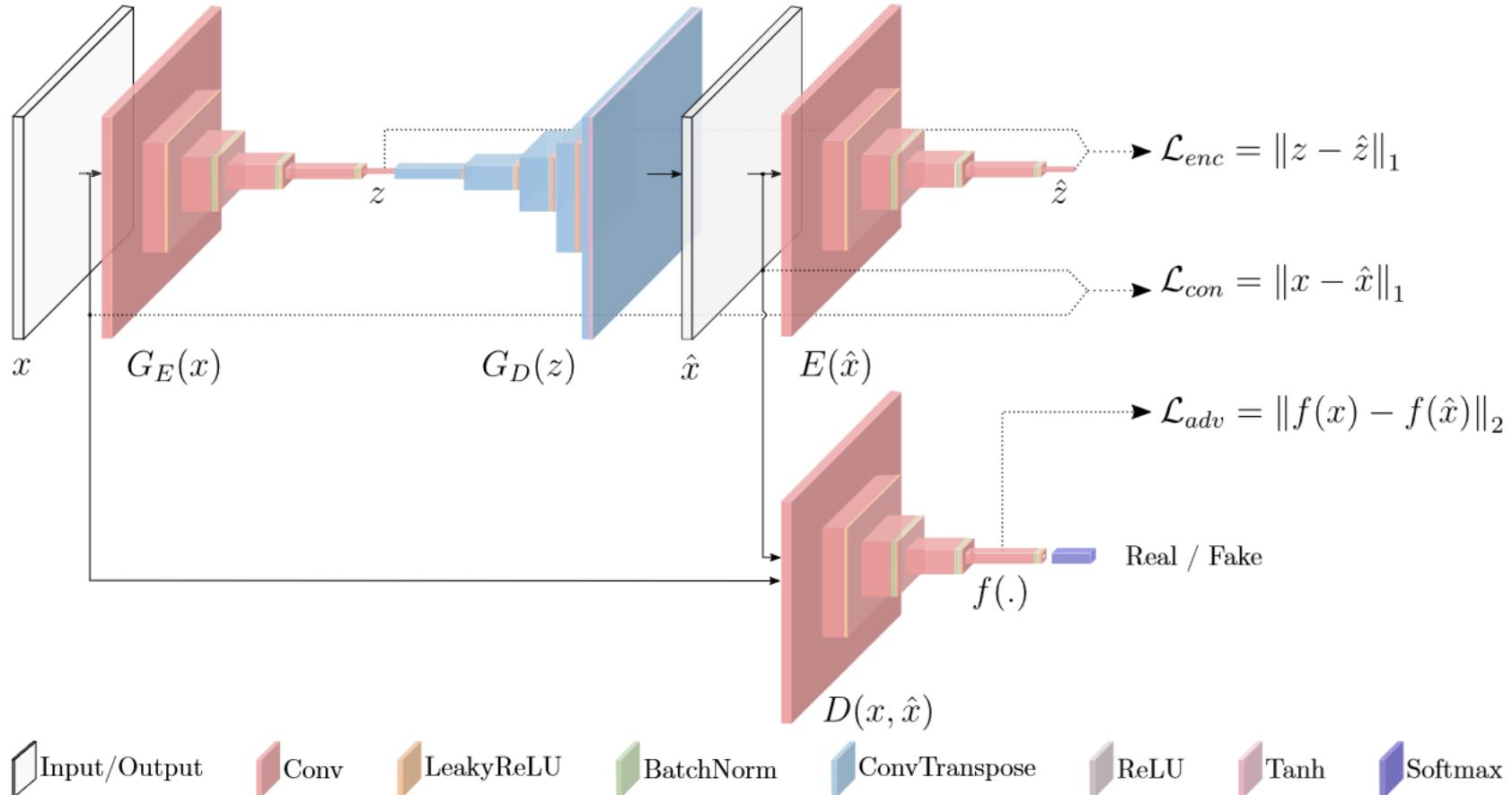


Figure 3. GANomaly architecture and loss functions from (Akcay et al., 2018).

	KDD		
	Precision	Recall	F1-Score
BiGAN/EGBAD	0.941174	0.956155	0.948605
GANomaly	0.830256	0.841112	0.835648

Table 3. The performances of BiGAN/EGBAD and GANomaly models on the KDD dataset.

Summary

- There are many ways of finding the inverse of the Generator
- EGBAD use BiGAN architecture to train the encoder.
- GANomaly use an encoder-decoder structured generator and align the encoder inside the generator with an external encoder, which can be viewed as a part of the discriminator.

Data Augmentation GAN

- Antreas Antoniou, Amos Storkey, Harrison Edwards, "Data Augmentation Generative Adversarial Networks", 2017
- Motamed, Saman, and Farzad Khalvati. "Inception Augmentation Generative Adversarial Network." arXiv preprint arXiv:2006.03622 (2020).

DAGAN

作用在 VGG-Face 的結果(Test Accuracy)如下

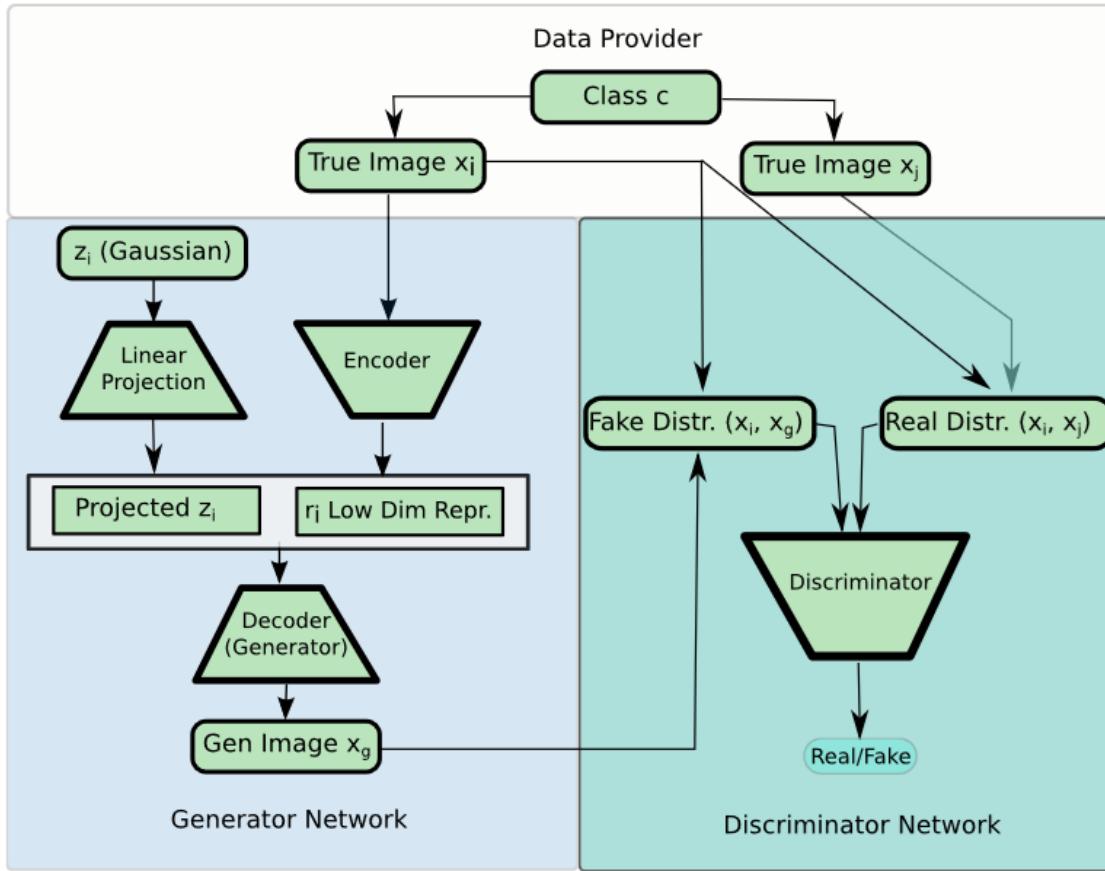
# Sample...	# Standard	# DAGAN
5	4.4%	12.6%
15	39.3%	42.9%
25	58%	58.5%

直接用 GAN 來 augmentation

利用 GAN 來學 augmentation.

- 紿定輸入圖片 x
- 先取得 representation $r=g(x)$,
- 然後利用生成模型 f 和 noise z 來生出 $x_2=f(z,r)$

這裡的生成模型可以視為 unsupervised meta-learning.



Inception-Augmentation GAN

架構類似 DAGAN，也是由輸入的圖片和 noise 產生新圖片

- 但包含較為先進的結構，像是 attention, inception blocks

輸入的圖片先

- 經過 CNN 和 attention 降低維度
- 和 noise concate
- 然後再進入後面的網路。

訓練 GAN 的時候：

- 輸入 x , 然後希望生出來的圖片盡量和 x 不同(以 SSIM 衡量)
- 但又符合原來的 distribution (conditional GAN 的意義下)。

當 GAN 訓練完後

- 利用 AnoGAN 的方式來判斷異常。

原始資料有 3265 張的肺部圖片。

- 在沒有 augmentation 的情形下, $AUC=0.83$
- 使用 IAGAN, $AUC=0.88$.

這個方法主要的優點是，3265 張的數量不多

- 在其他 GAN 失效的情形下，利用圖片輔助生成，可以讓 GAN 更容易生成。

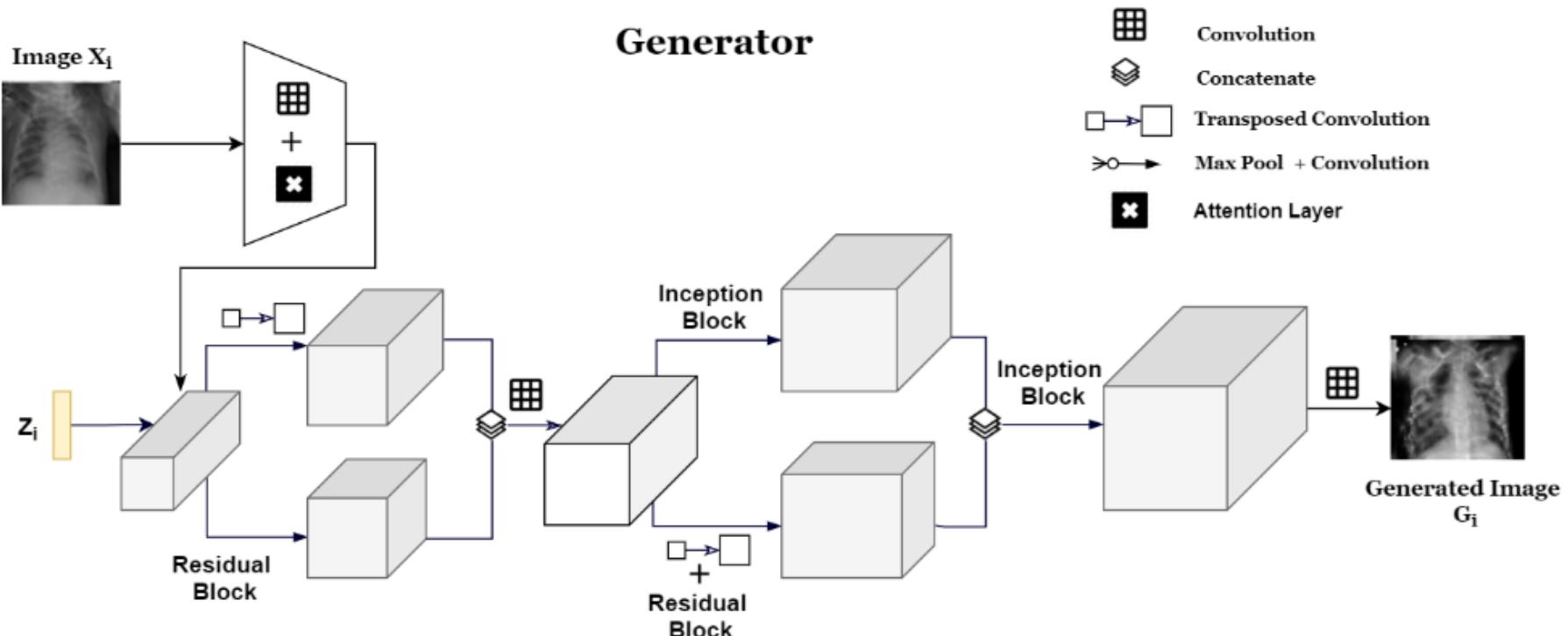


Figure 1: IAGAN’s Generator Architecture

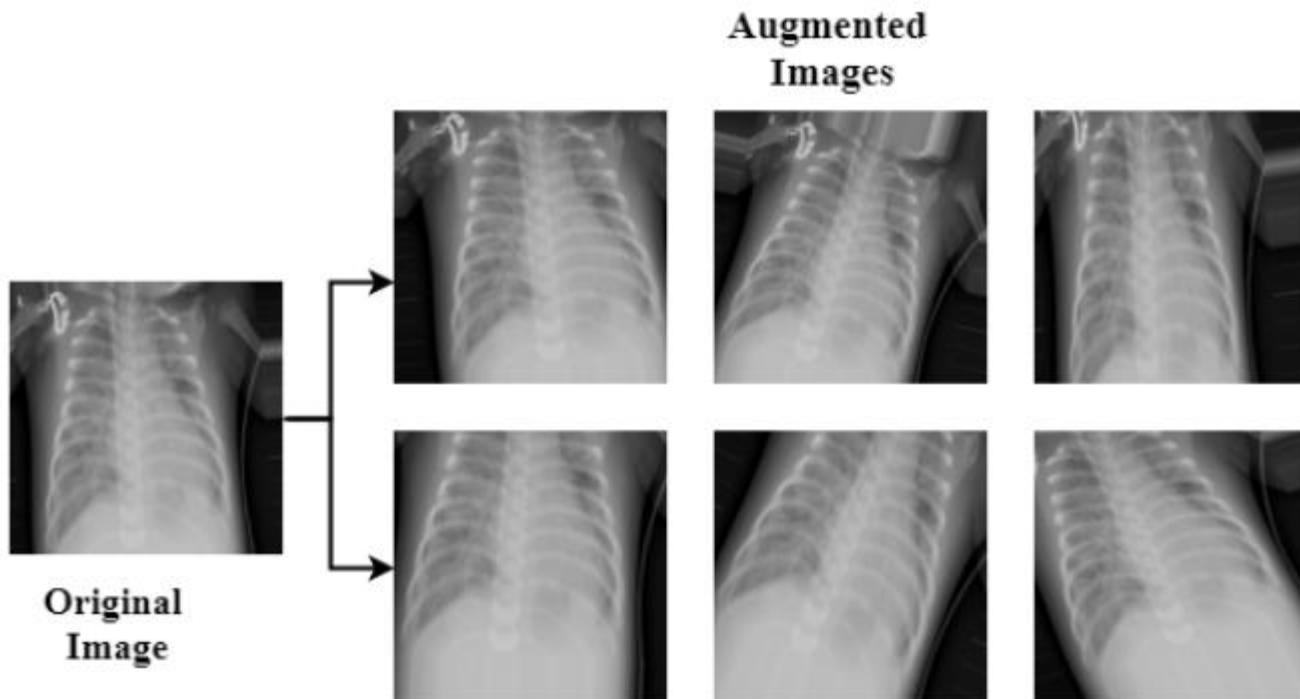


Figure 6: Traditional augmentation output sample

Summary

- 更多不同的 GAN 使用方式
- 利用圖到圖來
 - 降低訓練集數量的需求
 - 提升生成圖片的品質

