

# Image Anomaly Detection

Data Science Team

Alvin.lee

This presentation may contain confidential information and/or copyright material of kt NexR Data Science Team. and intended for the use of the NexR members only. DO NOT EXPRESS OR IMPLIED, IN THIS SUMMARY.

## 1. Anomaly Detection 개념 및 적용사례

## 2. Anomaly Detection 종류

- 학습유형에 따른 Anomaly Detection
- 비정상 데이터 종류에 따른 Anomaly Detection

## 3. Anomaly Detection 알고리즘 소개

- 딥러닝 생성모델(AutoEncoder, GAN)
- Anogan
- Fast-Anogan
- GANomaly
- Skip-GANomaly

## 4. Image Anomaly Detection 실습

- 건축 분야: 건물의 균열 데이터
- 의료 분야: 코로나 환자 흉부 X-ray 데이터
- 제조 분야: 물품의 흠집, 파손데이터

## 5. 결론 및 활용방안

## 1. Anomaly Detection 개념 및 적용사례

## 2. Anomaly Detection 종류

- 학습유형에 따른 Anomaly Detection
- 비정상 데이터 종류에 따른 Anomaly Detection

## 3. Anomaly Detection 알고리즘 소개

- 딥러닝 생성모델(AutoEncoder, GAN)
- Anogan
- Fast-Anogan
- GANomaly
- Skip-GANomaly

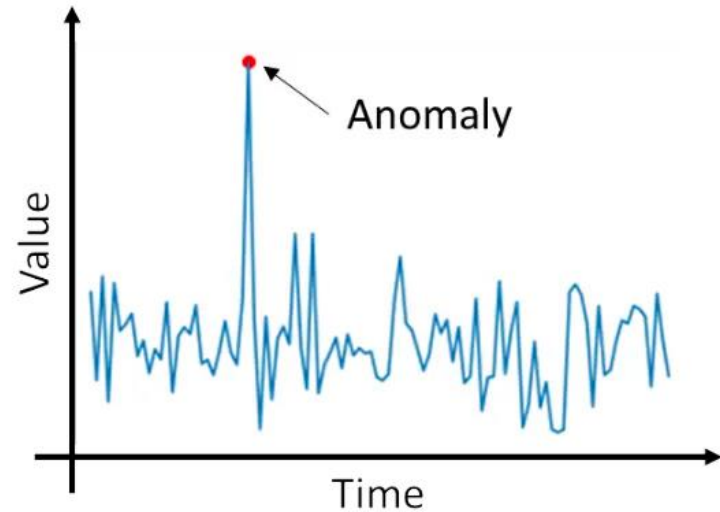
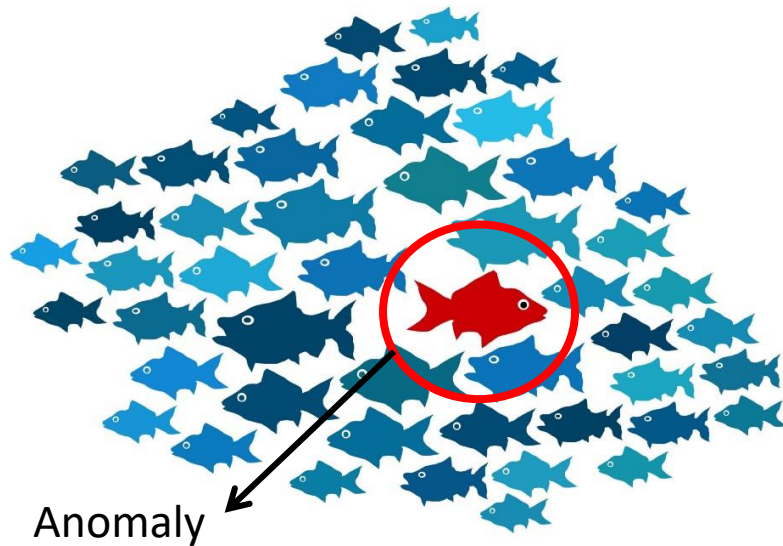
## 4. Image Anomaly Detection 실습

- 건축 분야: 건물의 균열 데이터
- 의료 분야: 코로나 환자 흉부 X-ray 데이터
- 제조 분야: 물품의 흠집, 파손데이터

## 5. 결론 및 활용방안

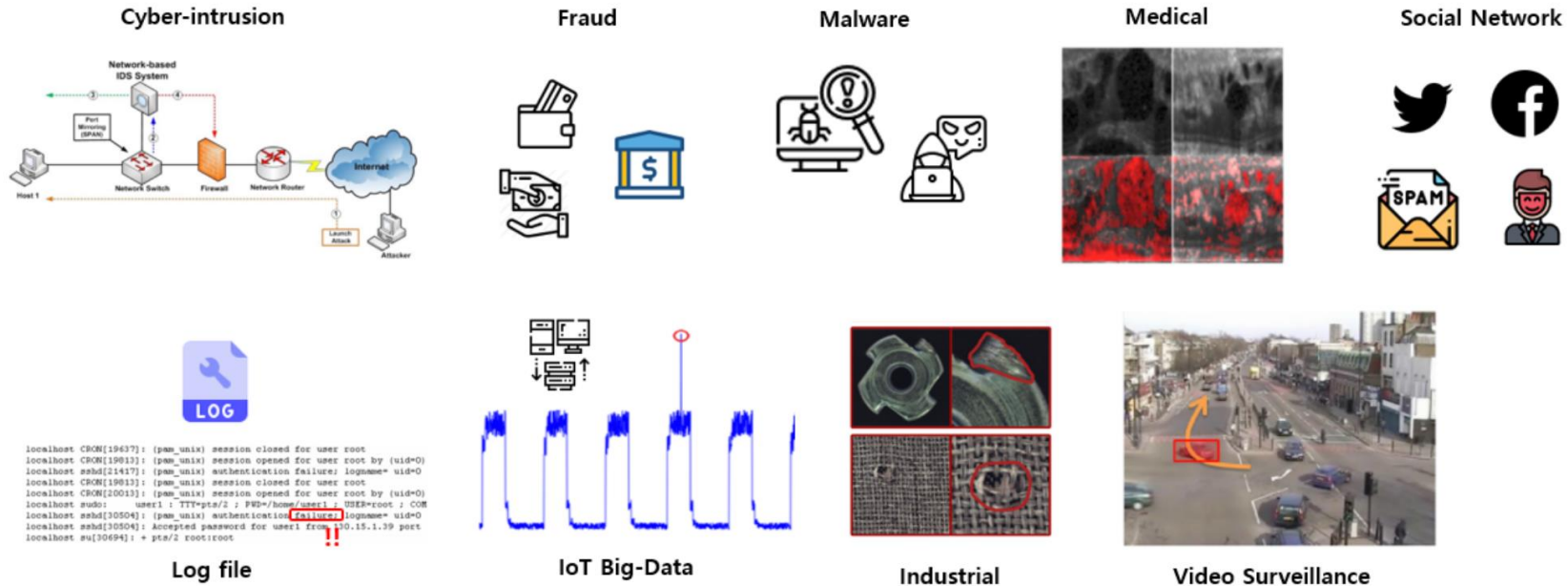
# 1-1. Anomaly Detection 개념

- **Anomaly Detection 이란?** 비정상(Abnormal) 데이터를 탐지하는 분석 방법
- **Anomaly Detection 목적**
  - 신용 카드 사기, 사이버 침입, 질병 탐지 등 실생활에서 접할 수 있는 피해를 방지하기 위해 사용



# 1-2. Anomaly Detection 적용 사례

## <Anomaly Detection의 적용 사례>



- 사기거래탐지(Fraud Detection): 보험, 신용, 금융 관련 데이터에서 불법 행위를 검출
- 바이러스탐지(Malware Detection): 컴퓨터 바이러스(Malware)를 검출
- 의료분야(Medical Anomaly Detection): 의료 영상, 뇌파 기록 등 의학 데이터에 대한 이상치 탐지
- 소셜 네트워크분야(Social Networks Anomaly Detection): Social Network 상의 이상치 탐지(Ex. 스팸)
- 제조업 분야(Industrial Anomaly Detection): 제조한 물품에 대한 불량품 혹은 이상 탐지

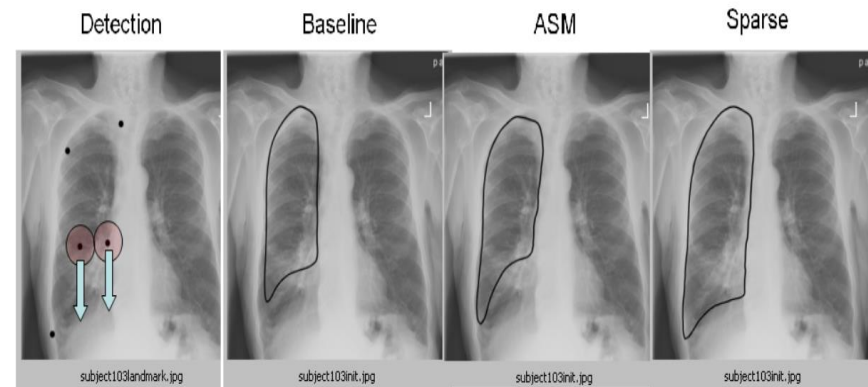
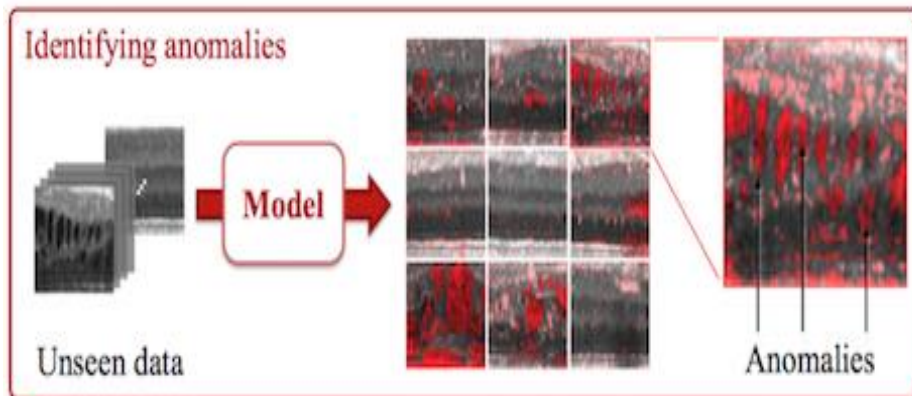
출처: DEEP LEARNING FOR ANOMALY DETECTION: A SURVEY

# 1-3. Image Anomaly Detection 적용 사례

- 의료 분야의 Anomaly Detection : 사람의 흉부에 종양, 용종 등의 질병을 탐지할 수 있는 모델 구축

## \*Image Anomaly Detection이 필요한 이유

- 1) 정확한 질병탐지의 필요성
- 2) 의료영상에 Labeling을 하는 것의 어려움
  - 의료분야에서는 학습을 하기 위해 의사의 수작업을 통해 질병이 찍힌 영상을 표시하여 데이터를 만들어야 함
- 3) 수집한 Label에 의존하는 문제
  - 탐지하지 못한 질병에 대해서는 Labeling을 할 수 없으며, Anomaly Score가 높은 이미지를 확인하여 새로운 질병을 탐지할 수 있음


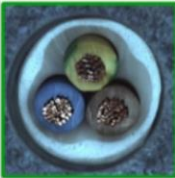

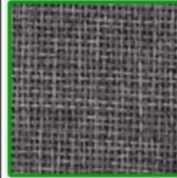
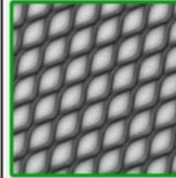

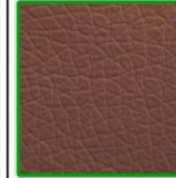
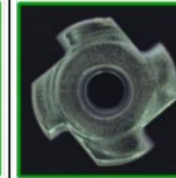

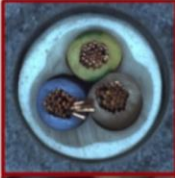

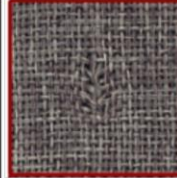
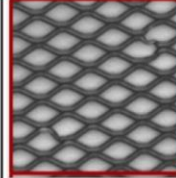
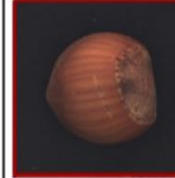
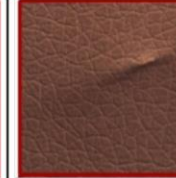



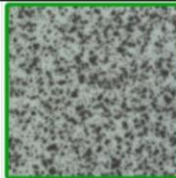

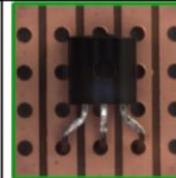

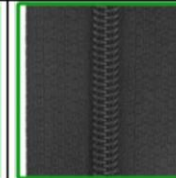


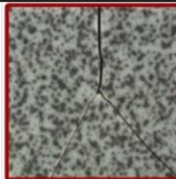

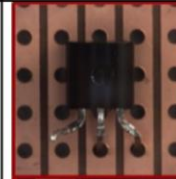

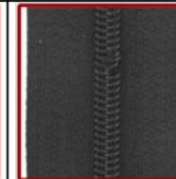


## <의료 분야의 Anomaly Detection 적용 사례>

출처:Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery

# 1-3. Image Anomaly Detection 적용 사례

- 제조업분야: 물품의 흠집, 깨짐 등의 현상으로 발생한 불량품 탐지( Ex. MVTec AD dataset)

Category	Bottle	Cable	Capsule	Carpet	Grid	Hazelnut	Leather	Metal nut
Normal Sample								
Anomaly Sample								
Category	Pill	Screw	Tile	Toothbrush	Transistor	Wood	Zipper	
Normal Sample								
Anomaly Sample								

출처: Anomaly Detection Neural Network with Dual Auto-Encoders GAN and Its Industrial Inspection Applications



## 1. Anomaly Detection 개념 및 적용사례

## 2. Anomaly Detection 종류

- 학습유형에 따른 Anomaly Detection
- 비정상 데이터 종류에 따른 Anomaly Detection

## 3. Anomaly Detection 알고리즘 소개

- 딥러닝 생성모델(AutoEncoder, GAN)
- Anogan
- Fast-Anogan
- GANomaly
- Skip-GANomaly

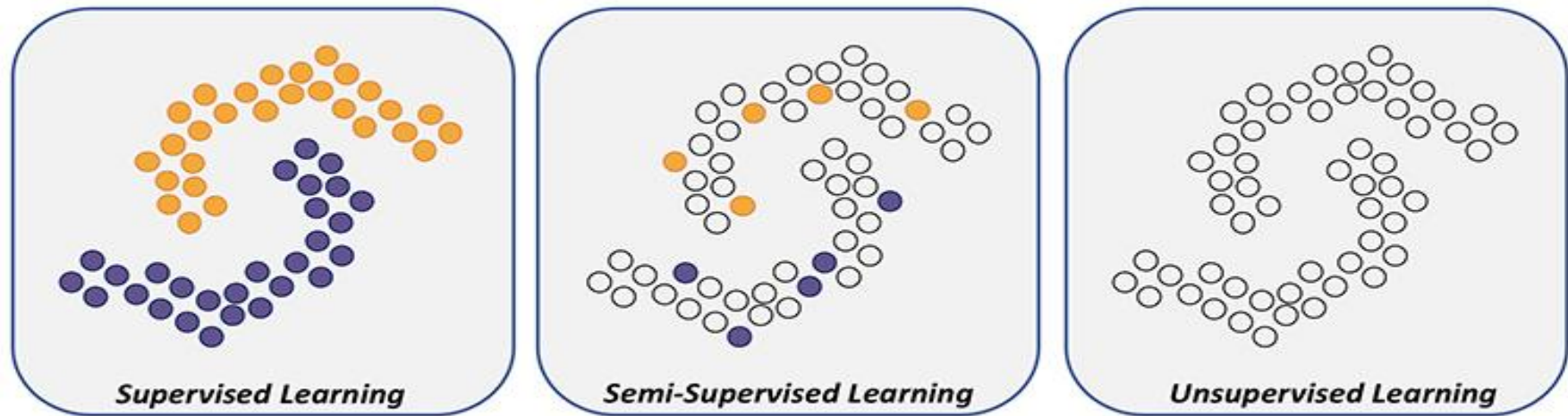
## 4. Image Anomaly Detection 실습

- 건축 분야: 건물의 균열 데이터
- 의료 분야: 코로나 환자 흉부 X-ray 데이터
- 제조 분야: 물품의 흠집, 파손데이터

## 5. 결론 및 활용방안



## 2-1. 학습 유형 분류



- 1) 지도학습(Supervised Learning): 데이터에 Label이 있으며 이를 통해 학습하여 새로운 데이터를 예측함 (Ex. 회귀, 분류)
- 2) 준지도학습(Semi-Supervised Learning): Label이 있는 데이터와 Label이 없는 데이터를 모두 사용
- 3) 비지도학습(Unsupervised Learning): 데이터에 Label이 없으며 비슷한 데이터들간에 특성을 파악 (Ex. 군집분석)

## 2-1. 학습 유형에 따른 Anomaly Detection 종류

### 1) Supervised Anomaly Detection

- 정상과 비정상의 데이터 및 Label을 모두 사용하여 학습하는 방법
- **장점:** 정상/비정상의 Label을 이용하기 때문에 분류정확도가 높은 편에 속함
- **단점:** 비정상 데이터를 수집하는 비용과 시간이 많이 들며 정상 데이터와 비정상 데이터의 Class 불균형 해소 필요

### 2) Semi-Supervised Anomaly Detection(One-class classification)

- **Label이 있는 데이터 중 정상 데이터만 이용하여** 모델을 학습하고 Label이 없는 테스트 데이터에 대해 비정상을 판단하는 방법  
(실습시에는 모델의 성능 검증을 위해 Label이 있는 테스트 데이터를 활용)
- 정상 데이터를 둘러싸는 경계를 설정하고 경계를 벗어날 경우 비정상으로 판단
- **장점:** 정상 데이터만으로 학습 가능
- **단점:** 일반적인 경우 Supervised Anomaly Detection 대비 정확도가 떨어짐

#### <One-class classification>

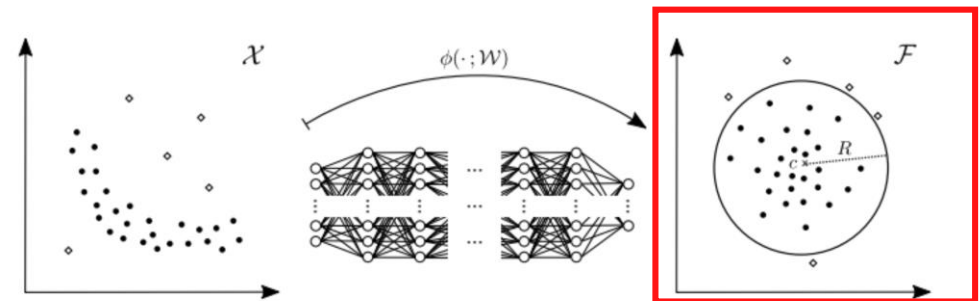


Figure 1. Deep SVDD learns a neural network transformation  $\phi(\cdot; W)$  with weights  $W$  from input space  $\mathcal{X} \subseteq \mathbb{R}^d$  to output space  $\mathcal{F} \subseteq \mathbb{R}^p$  that attempts to map most of the data network representations into a hypersphere characterized by center  $c$  and radius  $R$  of minimum volume. Mappings of normal examples fall within, whereas mappings of anomalies fall outside the hypersphere.

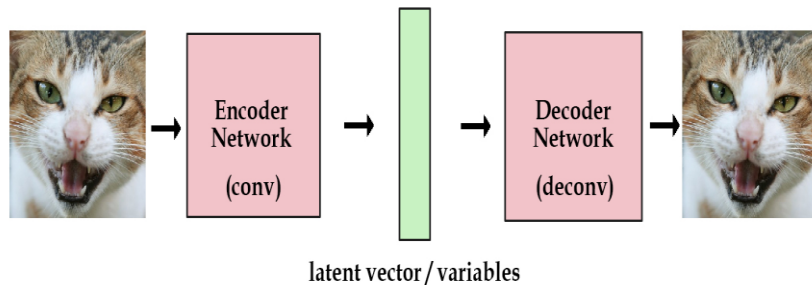
출처:Deep One-Class Classification

## 2-1. 학습 유형에 따른 Anomaly Detection 종류

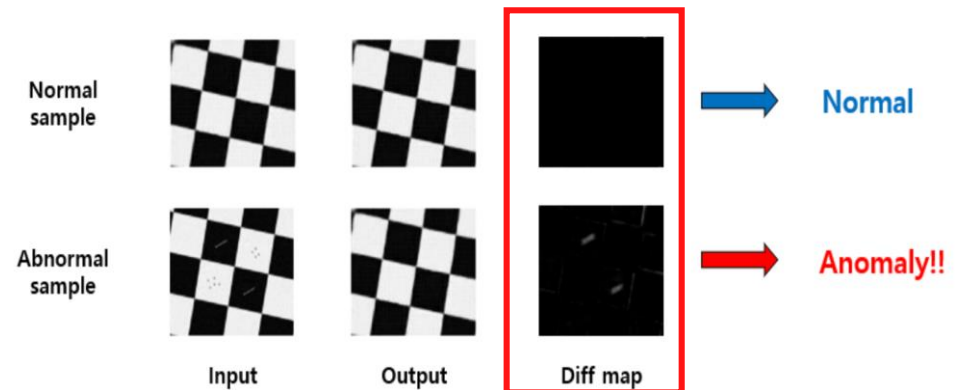
### 3) Unsupervised Anomaly Detection

- 정상/비정상의 데이터 모두 Label에 대한 정보가 없이 대부분의 데이터가 모두 정상이라고 가정하여 학습하는 방법
- PCA, Autoencoder 등으로 분석기법으로 차원을 축소하고 복원하는 과정을 통해 비정상 데이터 검출
- Autoencoder로 학습 후 정상 데이터를 입력하면 잘 복원 되지만, 비정상 데이터는 잘 복원이 되지 않는 원리를 사용
- **장점:** 데이터의 라벨이 필요 없음
- **단점:** 모델이 불안정함

<Autoencoder 구조>



<Autoencoder를 이용한 Anomaly Detection>



\*생성한 이미지와 실제 이미지의 차이로 Anomaly Score 계산

## 2-2. 비정상 데이터 유형에 따른 Anomaly Detection의 종류

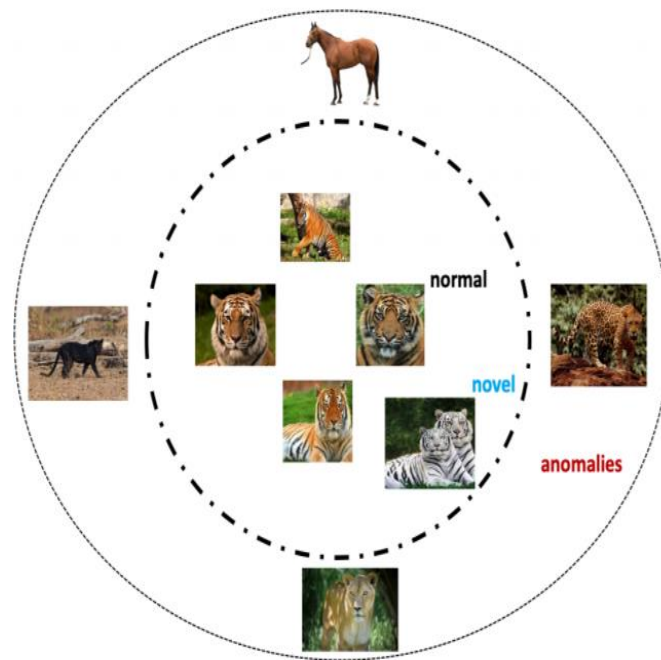
### 1) Novelty Detection (새로운 데이터 탐지 모델)

- 기존 데이터의 범주에 속하는 새로운 데이터를 탐지하는 모델
- 새로 입력된 데이터를 Novel sample 혹은 Unseen sample로 표현
- Ex) 호랑이 데이터 중에서 백호를 탐지하는 모델

### 2) Outlier Detection (이상치 탐지 모델)

- 기존 데이터와는 전혀 다른 데이터가 입력되는 경우
- 새로 입력된 데이터를 Outlier sample 혹은 Abnormal sample로 표현
- Ex) 호랑이 데이터 중에서 말, 표범 등 다른 종을 탐지하는 모델

#### <Novelty Sample VS Outlier Sample>



출처: DEEP LEARNING FOR ANOMALY DETECTION: A SURVEY

## 2-4. Anomaly Detection의 종류

### 1. 학습 유형에 따른 Anomaly Detection

학습유형	사용 데이터	데이터의 Label 존재여부
Supervised Anomaly Detection	학습데이터(Train Data)	O
	평가데이터(Test Data)	O
Semi - Supervised Anomaly Detection	학습데이터(Train Data)	O
	평가데이터(Test Data)	X
Unsupervised Anomaly Detection	학습데이터(Train Data)	X (모두 정상데이터라고 가정)
	평가데이터(Test Data)	X

### 2. 비정상 데이터 유형에 따른 Anomaly Detection

학습유형	비정상 데이터의 유형
Novelty Detection	기존 데이터의 범주에 속하는 새로운 데이터 탐지
Outlier Detection	기존 데이터와는 전혀 다른 유형의 데이터로 이상치로 판단할 수 있는 데이터 탐지

#### \*모델선정

- Semi-Supervised Anomaly Detection** 방식의 **Outlier Detection** 모델 구축
- GAN(Generative Adversarial Network)**을 활용하여 비정상 데이터 탐지

## 1. Anomaly Detection 개념 및 적용사례

## 2. Anomaly Detection 종류

- 학습유형에 따른 Anomaly Detection
- 비정상 데이터 종류에 따른 Anomaly Detection

## 3. Anomaly Detection 알고리즘 소개

- 딥러닝 생성모델(AutoEncoder, GAN)
- Anogan
- Fast-Anogan
- GANomaly
- Skip-GANomaly

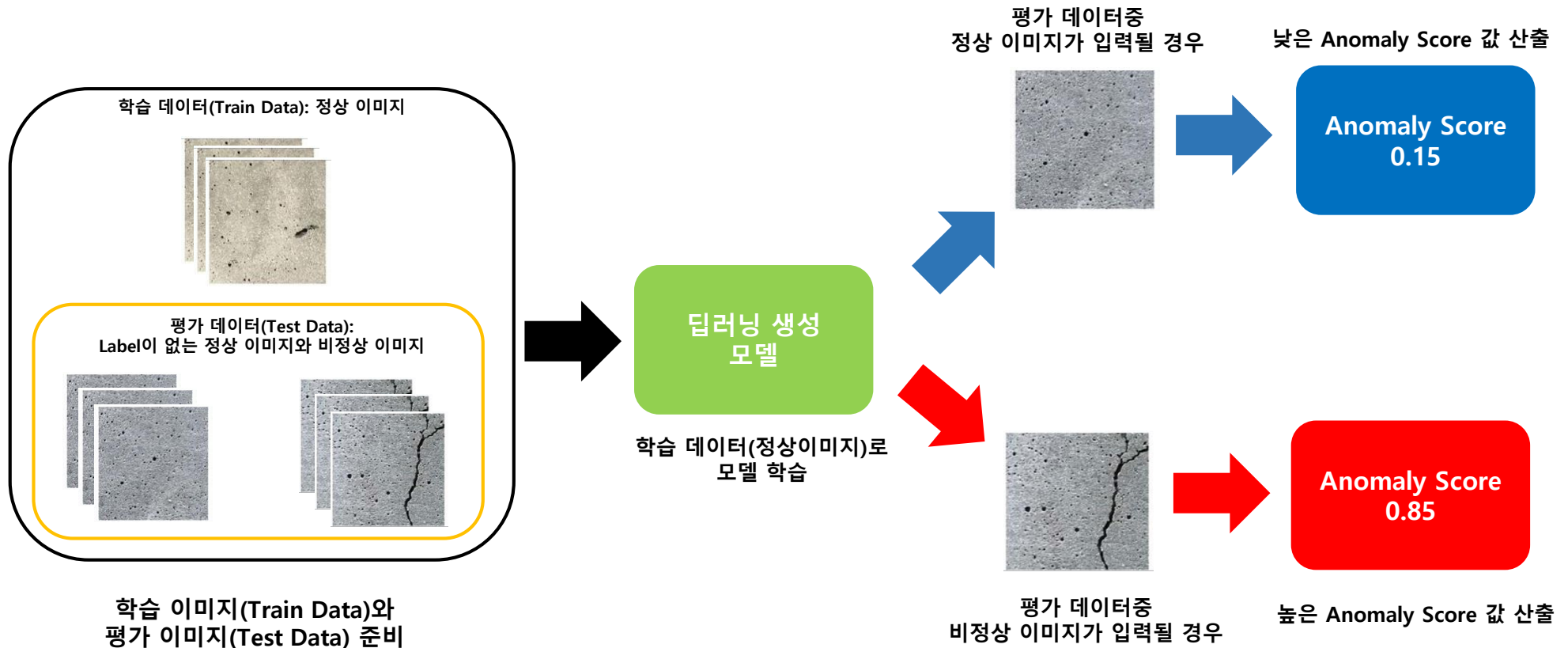
## 4. Image Anomaly Detection 실습

- 건축 분야: 건물의 균열 데이터
- 의료 분야: 코로나 환자 흉부 X-ray 데이터
- 제조 분야: 물품의 흠집, 파손데이터

## 5. 결론 및 활용방안

# 3-1. Semi-Supervised Anomaly Detection

## \*Semi-Supervised Anomaly Detection 모델 수행 과정





## 3-1. 딥러닝 생성모델

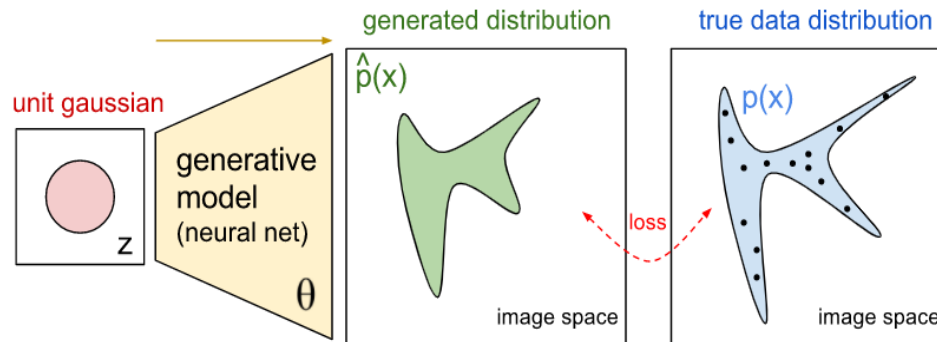
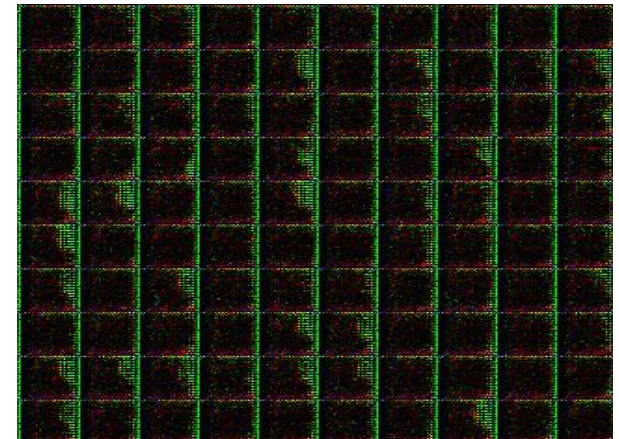
### \*딥러닝 생성모델(Deep Generative Model)

- 실제 데이터의 분포와 비슷한 분포를 가진 새로운 데이터를 생성하는 딥러닝 모델로, 대표적으로 **AutoEncoder**와 **GAN**이 있음
- 이미지 생성모델에서는 가상의 이미지 생성, 복원, 변환, 위조 등 다양한 분야에서 활용하고 있음

#### <AutoEncoder 이미지 생성>



#### <GAN 이미지 생성>

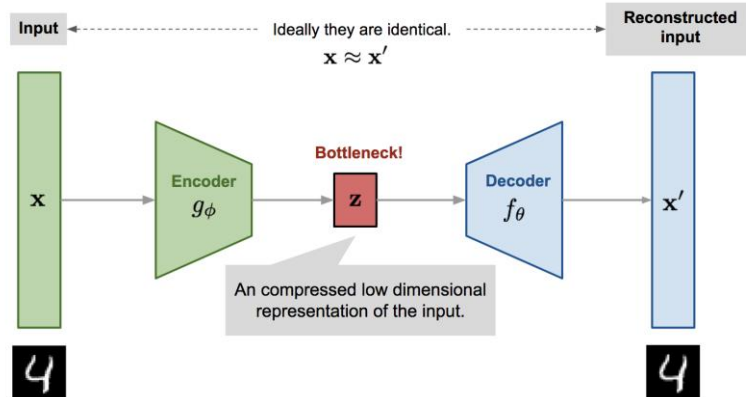


#### <딥러닝 생성모델>

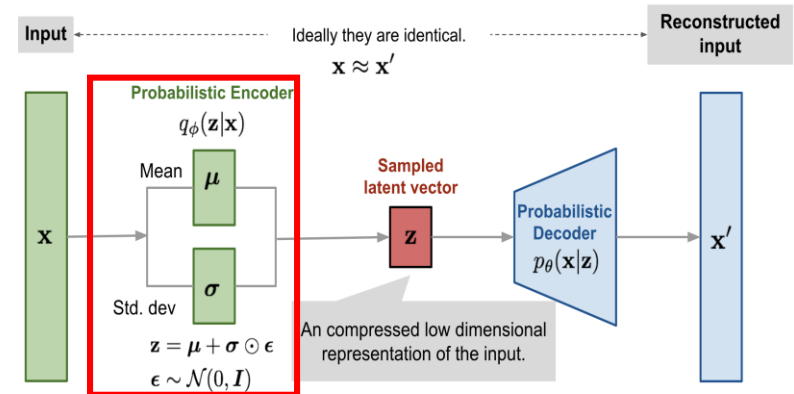
출처: <https://openai.com/blog/generative-models/>

# 3-1. AutoEncoder

## <AutoEncoder>



## <Variational AutoEncoder>



### \*AutoEncoder

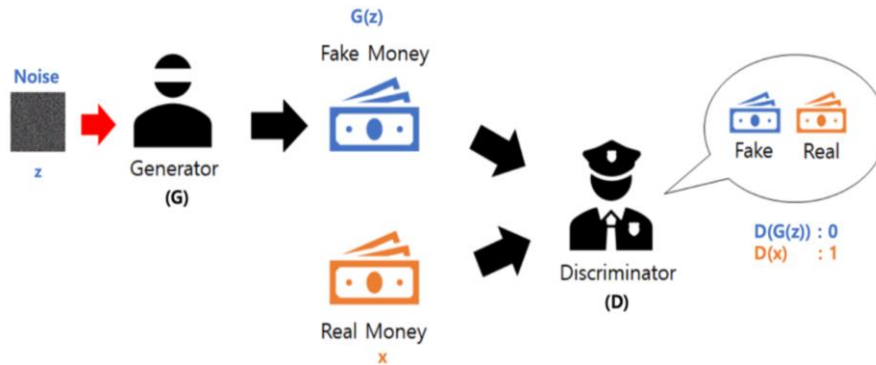
- 데이터를 Encoder를 통해 압축시키고 Decoder를 통해 복원하여 기존 데이터와 흡사한 데이터를 만드는 분석 기법
- Latent Space 안에서 각 데이터들의 특징 확인 가능
- 실제 이미지와 생성한 이미지의 MSE 차이를 줄이도록 학습하는 Reconstruction loss 사용

### \*Variational AutoEncoder(VAE)

- Encoder로 학습한 Latent space의 평균과 표준편차를 바탕으로 만든 다변량 정규 분포에서 랜덤하게 값을 뽑아 Decoder로 생성
- Latent Space가 다변량 정규분포를 따른다고 가정하며, 일반 AutoEncoder보다 다양한 데이터 생성 가능

## 3-1. GAN

### <GAN의 구조>



**\*GAN: 실제 데이터의 확률 분포와 비슷한 분포의 새로운 데이터를 만드는 방법**

1) 구조

-Generator(생성자)와 Discriminator(구별자)로 구성

2) 목표

-Generator의 목표: 구별자를 속일 수 있는 데이터 생성

-Discriminator의 목표: 생성 데이터를 가짜데이터로 판별

\*수식

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

생성 데이터의 분포

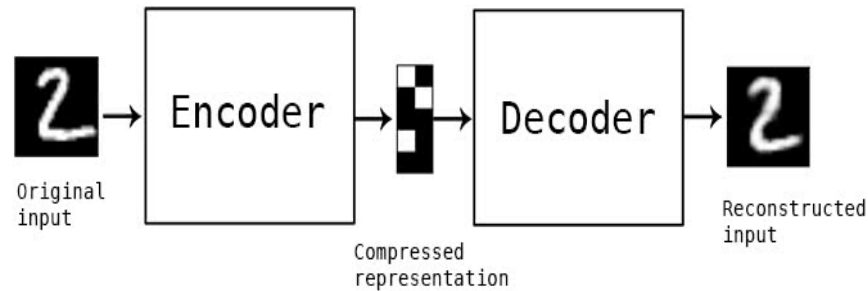
실제 데이터의 분포

\*Generator:  $V(D, G)$ 가 최소가 되도록 학습

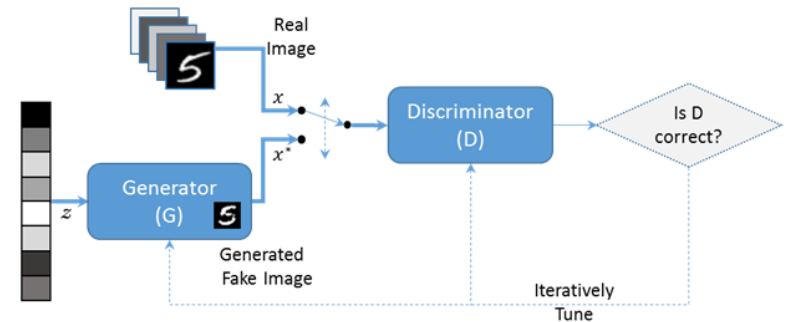
\*Discriminator:  $V(D, G)$ 가 최대가 되도록 학습

## 3-1. AutoEncoder VS GAN

<AutoEncoder >



<GAN>



### \*공통점

- 1) 딥러닝 기반의 생성모델
- 2) 모델 구조가 자유로움

- AutoEncoder의 Encoder, Decoder 및 GAN의 Generator와 Discriminator에 다양한 Architecture 사용 가능

Ex) GAN- Generator: LSTM, Discriminator: CNN

### \*차이점

- 1) AutoEncoder는 Encoder와 Decoder가 연결된 구조로 학습하는 반면, GAN은 Generator와 Discriminator가 번 갈아가며 학습
  - **AutoEncoder**: Encoder  $\rightarrow$  latent space( $z$ )  $\rightarrow$  Decoder(연결된 구조)
  - **GAN**: latent space( $z$ )  $\rightarrow$  Generator, Real Sample, fake sample  $\rightarrow$  Discriminator(분리된 구조)

## 3-2. Image Anomaly Detection 모델 종류

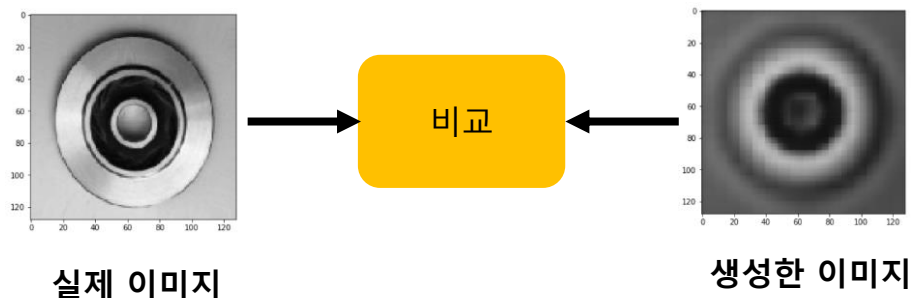
Model	모델 구조	설명	연도
AnoGAN	GAN Based	-GAN에서 정상데이터로만 학습	2017.03
EGBAD	GAN Based	-Anogan과 BiGAN을 합친 모델 -Anogan의 실행시간의 성능 이슈 해결	2018.06
GANomaly	GAN Based+ AutoEncoder	-GAN의 Generator에 AutoEncoder 사용	2018.02
F-AnoGAN	GAN Based + AutoEncoder	-GAN에서 AutoEncoder의 Encoder 추가 -Anogan의 실행시간 성능 개선	2019.01
Skip-GANomaly	GAN Based + AutoEncoder	-GANomaly Model에서 Skip-Connection 추가	2019.01
MemAE	AutoEncoder Based	-변형된 AutoEncoder 사용	2019.08
Adversial dual Autoencoder	AutoEncoder Based + GAN	-AutoEncoder에 GAN의 Discriminator 추가	2020.02

### \*Image Anomaly Detection 모델

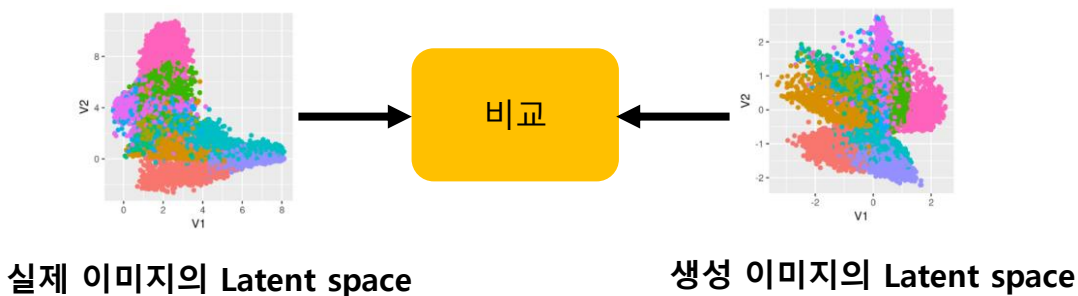
- AutoEncoder 또는 GAN의 딥러닝 생성 모형 사용
- GAN 또는 AutoEncoder 단일모델 보다는 GAN과 AutoEncoder의 구조를 합쳐 단점을 보완
- 알고리즘 소개 모델: AnoGAN, GANomaly, F-AnoGAN, Skip-GANomaly(            표시)
- 실습 구현 모델: GANomaly, F-AnoGAN

## 3-2. Image Anomaly Detection by GAN

### < 실제 이미지와 생성 이미지의 차이계산 >



### < 실제 이미지와 생성이미지의 Latent Space 차이계산 >



### \* GAN을 사용한 Anomaly Detection의 학습목표

- 1) 정상 데이터의 이미지와 흡사한 이미지 생성
- 2) 모델에서 정상 데이터의 Latent space의 분포를 따르는 최적의 Latent space를 찾는 것

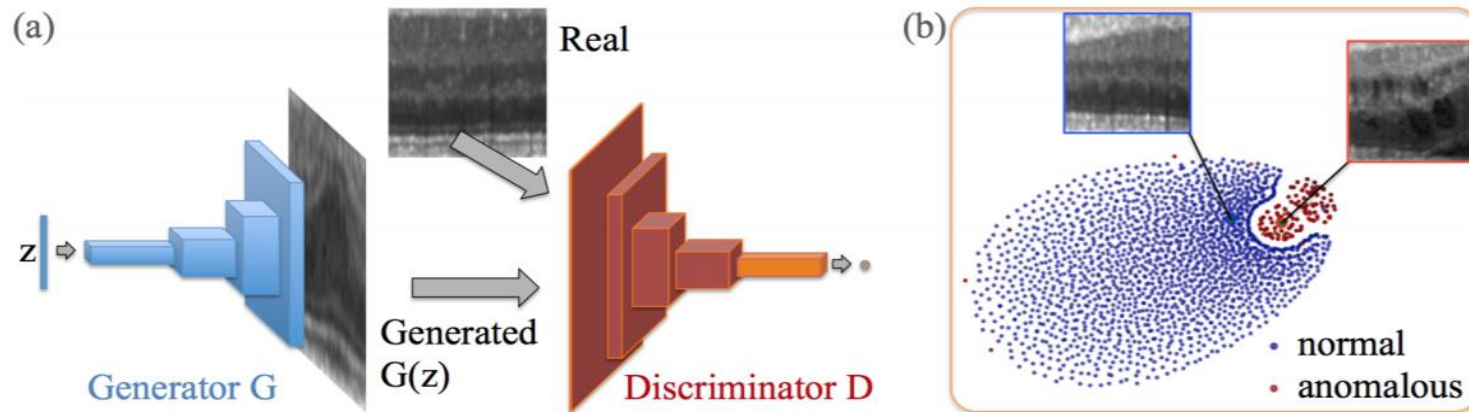
### \* GAN을 사용한 Anomaly Detection의 학습단계

- 1) 정상 데이터만을 이용하여 GAN을 학습
- 2) 모델의 Latent Space와 정상데이터의 Latent Space가 같아지도록 학습
- 3) 비정상 데이터와 정상 데이터를 테스트하여 Anomaly Score 산출

### \* 평가지표

- 1) 정성적 지표  
생성한 이미지와 실제이미지의 그림 비교(눈으로 비교)
- 2) 정량적 지표  
Anomaly Score를 바탕으로 ROC 커브, Recall, Precision 계산

### 3-3. AnoGAN



**Fig. 2.** (a) Deep convolutional generative adversarial network. (b) t-SNE embedding of normal (blue) and anomalous (red) images on the feature representation of the last convolution layer (orange in (a)) of the discriminator.

#### \*AnoGAN

- 정상(normal) 데이터만을 이용하여 학습하는 GAN 모형
- CNN(Convolution Neural Network)을 사용하며 latent space의 새로운 데이터에 대해 Anomaly Score 계산
- AnoGAN 학습 완료시 정상 데이터는 Generator의 latent space의 공간 안에 존재하며 비정상 데이터는 존재하지 않을 것임
- 정상이미지와 유사한 최적의 Latent space를 찾고, Anomaly score를 통해 비정상 데이터를 탐지

(최적의 Latent Space를 찾기 위해서는  $X \rightarrow \text{latent space } Z$ 의 과정이 필요하지만, GAN은  $\text{latent space } Z \rightarrow X$ 로 학습하기 때문에 이를 해결하는 것이 AnoGAN의 핵심 알고리즘이라고 할 수 있음)

출처: Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery



## 3-3. AnoGAN

---

### 1) Residual Loss

$$\mathcal{L}_R(\mathbf{z}_\gamma) = \sum |\mathbf{x} - G(\mathbf{z}_\gamma)|.$$

- $\mathbf{x}$  : 실제 이미지
- $G(\mathbf{z})$  : 생성한 이미지

### 2) Discriminator Loss

$$\mathcal{L}_D(\mathbf{z}_\gamma) = \sum |\mathbf{f}(\mathbf{x}) - \mathbf{f}(G(\mathbf{z}_\gamma))|,$$

- $\mathbf{F}$  : discriminator의 중간층에 있는 activations 값 (Feature Mapping)
- $\mathbf{z}$  : random sampling 값
- $\gamma$  :  $\mathbf{z}$ 의 횟수

### 3) Total Loss

$$\mathcal{L}(\mathbf{z}_\gamma) = (1 - \lambda) \cdot \mathcal{L}_R(\mathbf{z}_\gamma) + \lambda \cdot \mathcal{L}_D(\mathbf{z}_\gamma).$$

- $\lambda$  : 가중치 (default: 0.1)

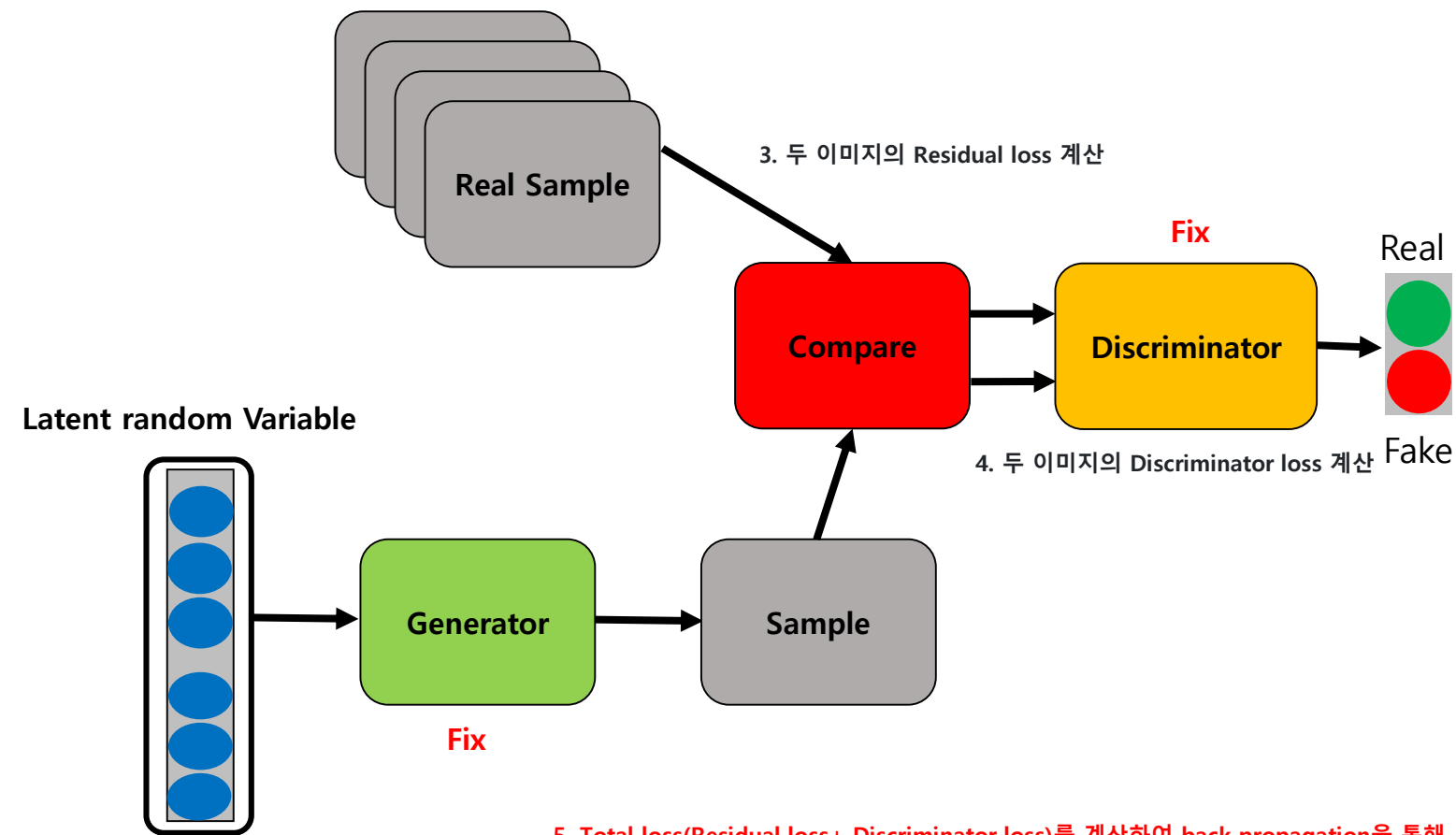
### \*AnoGAN은 2개 Loss의 가중치를 사용

- 1) Residual Loss:  $G(\mathbf{z})$  (생성 이미지 값)와  $\mathbf{x}$  (실제 이미지 값)의 차이
- 2) Discriminator Loss: 최적의 Latent space를 찾기 위한 loss (GAN의 discriminator의 loss가 아님)
- 3) Overall loss: Residual loss와 discriminator loss의 가중 합

### 3-3. AnoGAN

#### . AnoGAN의 학습과정

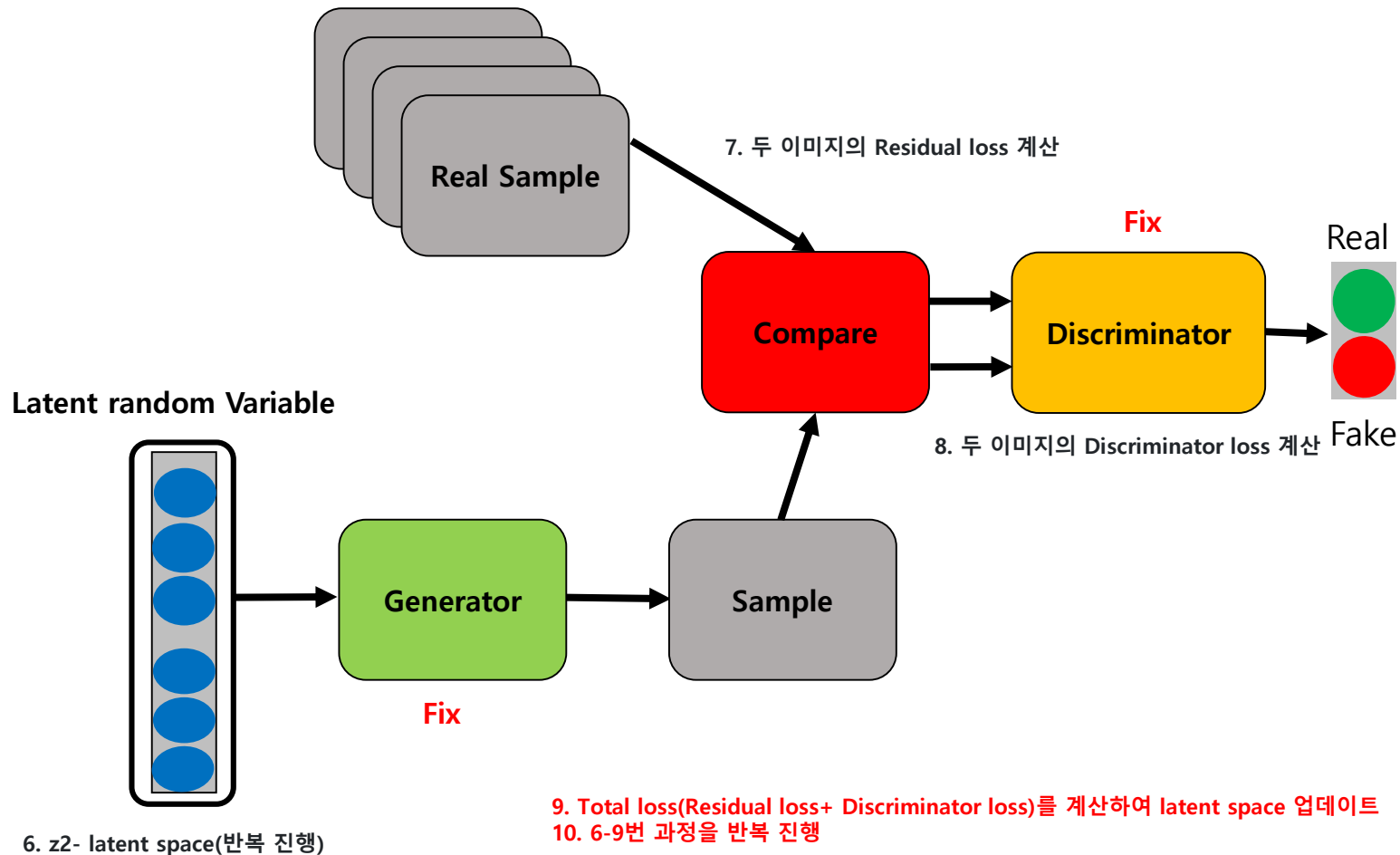
##### 1. 정상 데이터로 먼저 GAN의 Generator와 Discriminator 학습



##### 2. Random Sampling으로 z1 생성

### 3-3. AnoGAN

#### . AnoGAN의 학습과정



### 3-3. AnoGAN

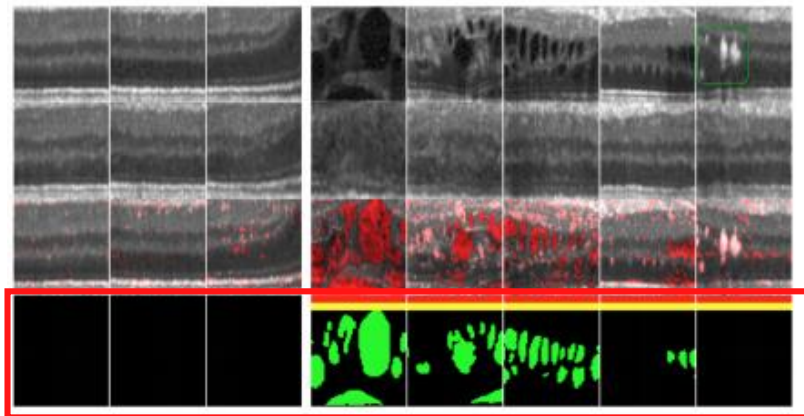
#### \* Anogan의 평가지표

##### <Anomaly Score>

$$\begin{aligned} A(\mathbf{x}) &= (1 - \lambda) \cdot R(\mathbf{x}) + \lambda \cdot D(\mathbf{x}) \\ &= (1 - \lambda) \cdot L_R(\mathbf{z}_\Gamma) + \lambda \cdot L_D(\mathbf{z}_\Gamma) \end{aligned}$$

- $A(x)$  : Anomaly Score(score 값이 클수록 비정상 데이터)
- $R(x)$ : Residual loss와 동일
- $D(x)$  : Discriminator loss 부분과 동일함
- $\Gamma$  : 마지막의 update iteration

##### <생성한 이미지와 실제 이미지 비교>

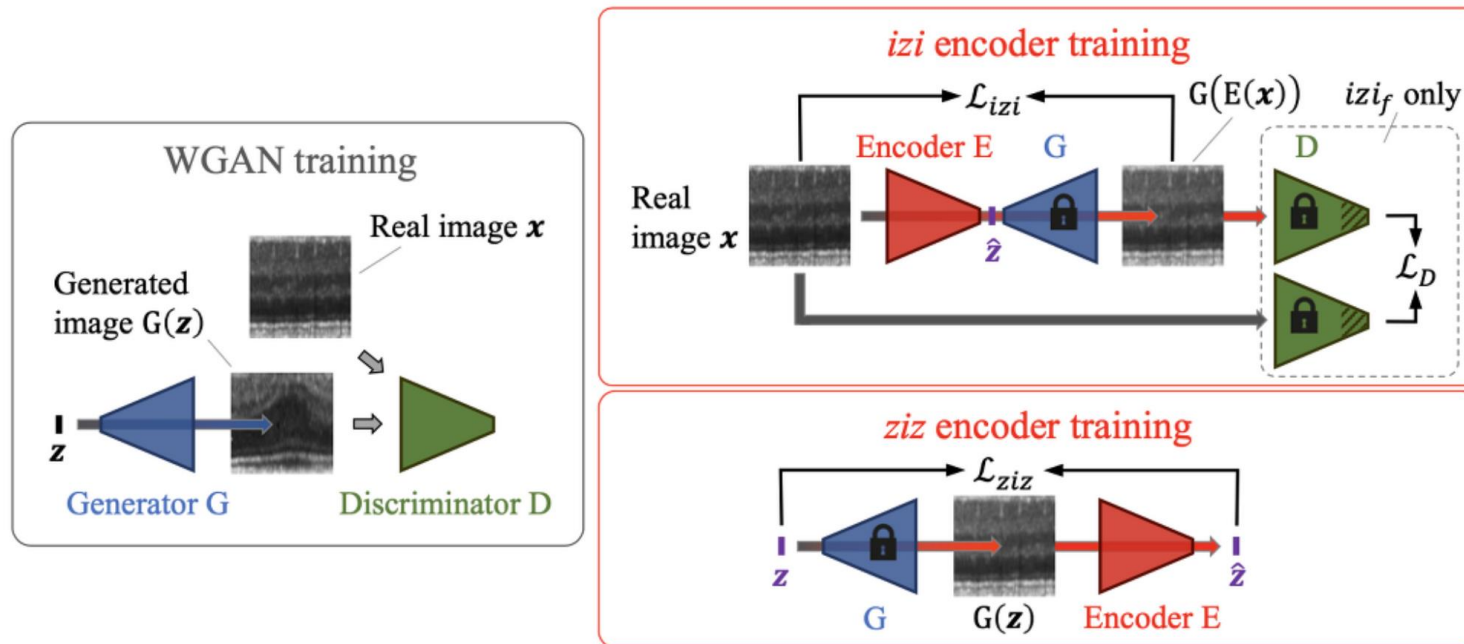


정상 데이터

비정상 데이터

- **정성적 지표**
  - 생성한 이미지와 실제이미지의 그림을 통한 비교(눈으로 비교)
  - 빨간색 박스가 생성한 이미지와 실제 이미지의 차이를 나타냄 (비정상 데이터의 경우 초록색 부분이 많음)
- **정량적 지표**
  - Anomaly Score 혹은  $D(x)$ 의 값을 바탕으로 ROC 커브 , Recall, Precision 계산

### 3-4. F-AnoGAN

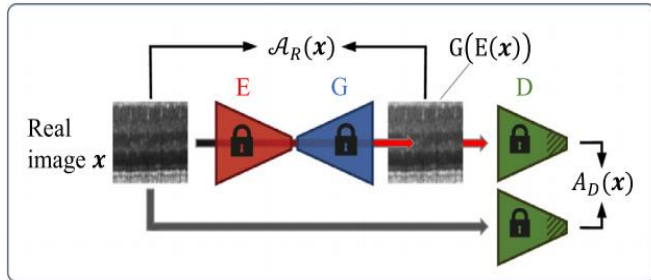


#### \*F-AnoGAN(Fast-AnoGAN)

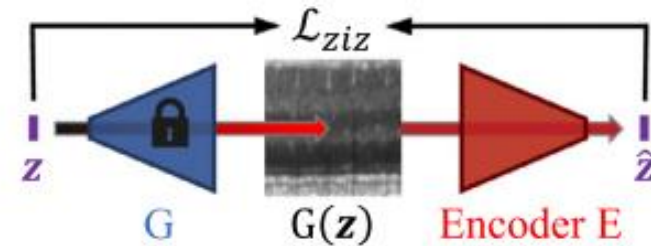
- Anogan의 단점: 이미지가 클수록 시간이 배로 걸리고 학습이 잘되지 않음
- F-Anogan은 AutoEncoder의 Encoder를 사용하여 정상데이터의 Latent space를 학습함
- Image-Z(latent Space)- Image 구조와 Z(latent Space)- Image- Z(latent Space) 구조로 이루어져 있음
- Anogan과 같은 방식으로 정상데이터로 wGAN으로 먼저 학습하고 Encoder를 그 다음으로 따로 학습함

### 3-4. F-AnoGAN

< Image- Z(latent Space) – Image 구조 >



< Z(latent Space)- Image – Z(latent Space) 구조 >



비교	izi 구조	ziz 구조
Loss함수	$\mathcal{L}_{izi}(\mathbf{x}) = \frac{1}{n} \ \mathbf{x} - G(E(\mathbf{x}))\ ^2$ <p><math>\mathbf{x}</math> : 실제 이미지  <math>G(E(\mathbf{x}))</math> : 생성한 이미지</p>	$\mathcal{L}_{ziz}(\mathbf{z}) = \frac{1}{d} \ \mathbf{z} - E(G(\mathbf{z}))\ ^2$ <p><math>\mathbf{z}</math> : 실제 이미지의 Z  <math>G(E(\mathbf{x}))</math> : 생성한 이미지의 z</p>
학습방법	<p>- 실제 이미지와 생성한 이미지의 차이 계산                      (실제 이미지(x)가 Encoder를 거쳐 latent Z가 되고, 이 Z를 기존에 학습된 Generator에 넣어 이미지(G(E(x))) 생성)                      - Loss: 실제 이미지와 생성한 이미지의 MSE</p>	<p>- 실제 이미지의 Latent Space와 생성이미지의 Latent Space 차이계산                      (Random Sampling한 z가 기존에 학습된 Generator를 거쳐 이미지를 만들고 이를 Encoder를 거쳐 latent E(G(z)) 생성)                      - Loss: 실제 이미지의 Latent z와 생성한 이미지의 Latent E(G(Z))의 MSE</p>
한계점	실제 이미지가 latent space 어디에 존재하는 지 알 수 없음	테스트(새로운) 이미지에 대해 잘 Mapping 하지 못함
최종 Loss 함수	$\mathcal{L}_{izif}(\mathbf{x}) = \frac{1}{n} \cdot \ \mathbf{x} - G(E(\mathbf{x}))\ ^2 + \frac{\kappa}{n_d} \cdot \ f(\mathbf{x}) - f(G(E(\mathbf{x})))\ ^2$ <p>*izi 구조와 ziz 구조의 Loss 함수를 모두 반영</p>	

## 3-5. GANomaly

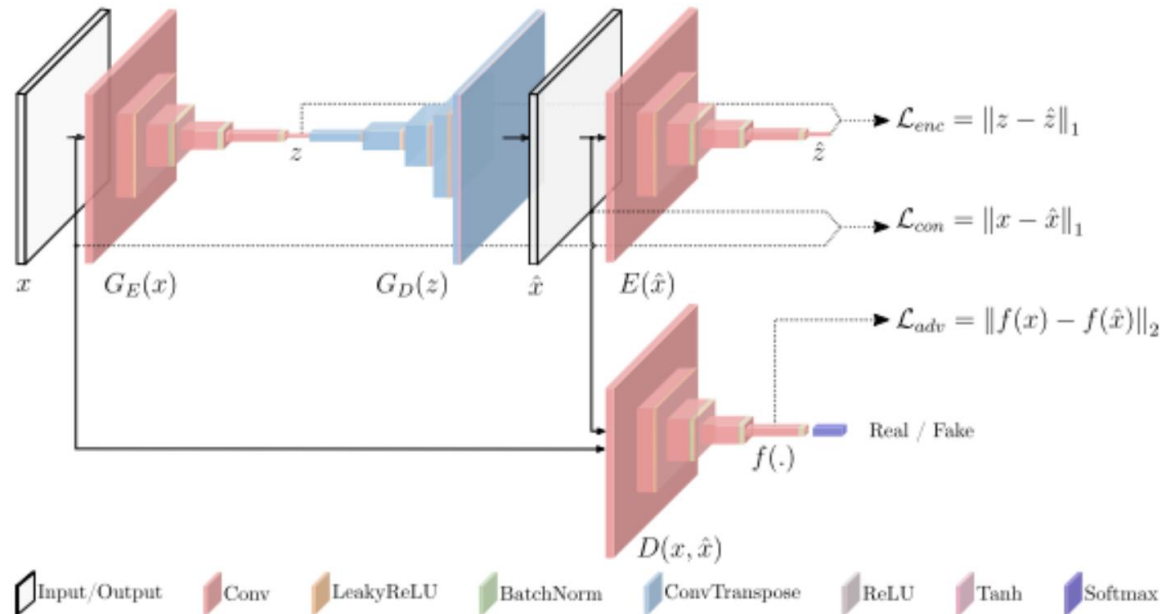


Figure 3. GANomaly architecture and loss functions from (Akçay et al., 2018).

### \*GANomaly

- Autoencoder와 GAN을 합친 Anomaly Detection 분석 방법
- Generator 부분에 AutoEncoder가 들어가 있는 것이 특징이며 Encoder 2개, Decoder 1개로 이루어져있음

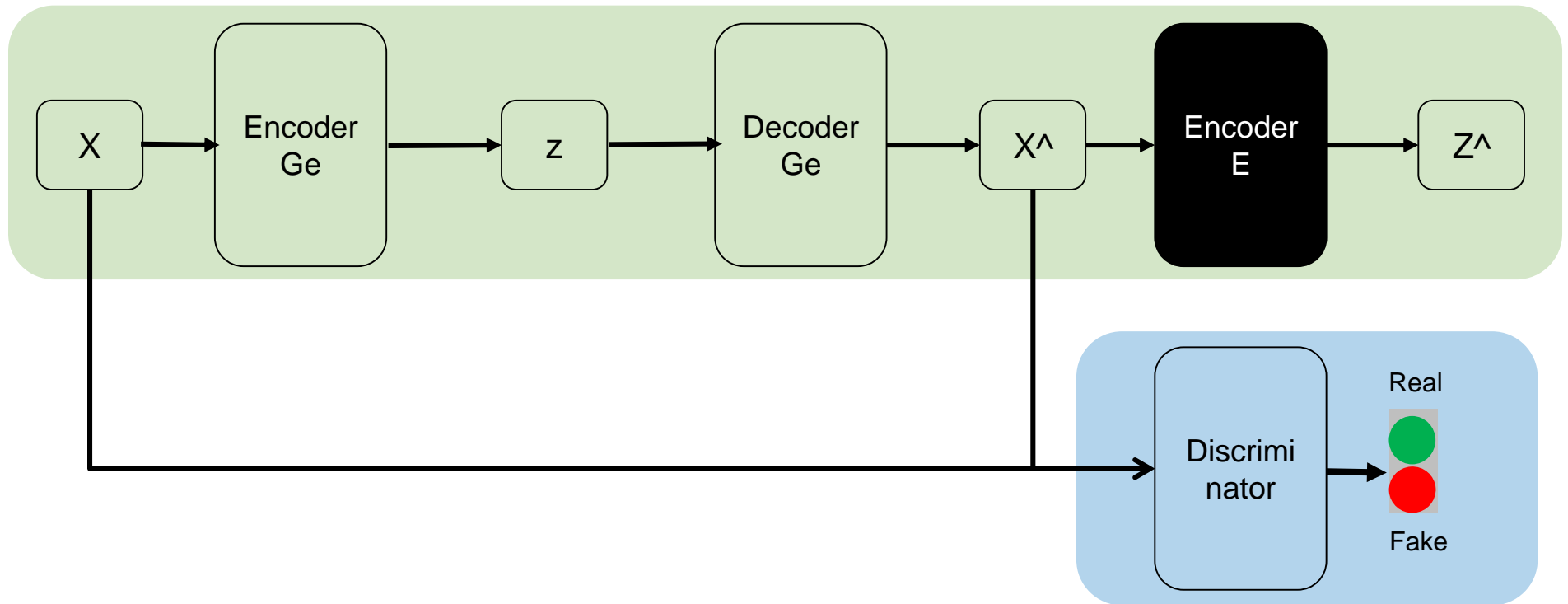
### \* GANomaly에 Autoencoder 사용하는 이유

- 정상 데이터를 다시 생성함으로써, Generator는 정상데이터에 더욱 집중할 수 있음
- Image  $\rightarrow$  Latent Space  $z$  가능

출처: GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training



## 3-5. GANomaly



### \* Generator Network

- Encoder  $G_e$ , Decoder  $G_d$ , Encoder  $E$ 로 구성되어있으며 Encoder  $G_e$ 와 Encoder  $E$ 의 구조는 동일

### \* Discriminator Network

- 일반적인 GAN의 Discriminator와 동일

## 3-5. GANomaly

- GANomaly는 3가지의 Loss를 사용

Loss 유형	수식	Loss 설명
Adversarial Loss	$\mathcal{L}_{adv} = \mathbb{E}_{x \sim p_{\mathbf{X}}} \ f(x) - \mathbb{E}_{x \sim p_{\mathbf{X}}} f(G(x))\ _2$	Discriminator의 손실함수
Contextual Loss	$\mathcal{L}_{con} = \mathbb{E}_{x \sim p_{\mathbf{X}}} \ x - G(x)\ _1$	생성한 이미지와 실제 이미지의 차이 계산
Encoder Loss	$\mathcal{L}_{enc} = \mathbb{E}_{x \sim p_{\mathbf{X}}} \ G_E(x) - E(G(x))\ _2$	실제 이미지와 생성한 이미지의 Latent space간의 차이 계산
total loss	$\mathcal{L} = w_{adv}\mathcal{L}_{adv} + w_{con}\mathcal{L}_{con} + w_{enc}\mathcal{L}_{enc}$	-앞서 언급한 3개의 loss 합 -가중치를 통해 각 loss의 중요도 조절 가능

### \*Anomaly Score

- latent space 간에 Anomaly Score 계산
- 개별적인 anomaly score(S)를 모으고 Min-max scaling을 적용하여 0-1 사이 값을 반환
- F-Anogan과 동일한 평가지표 활용하였으며, 비정상데이터와 정상데이터의 분포 확인

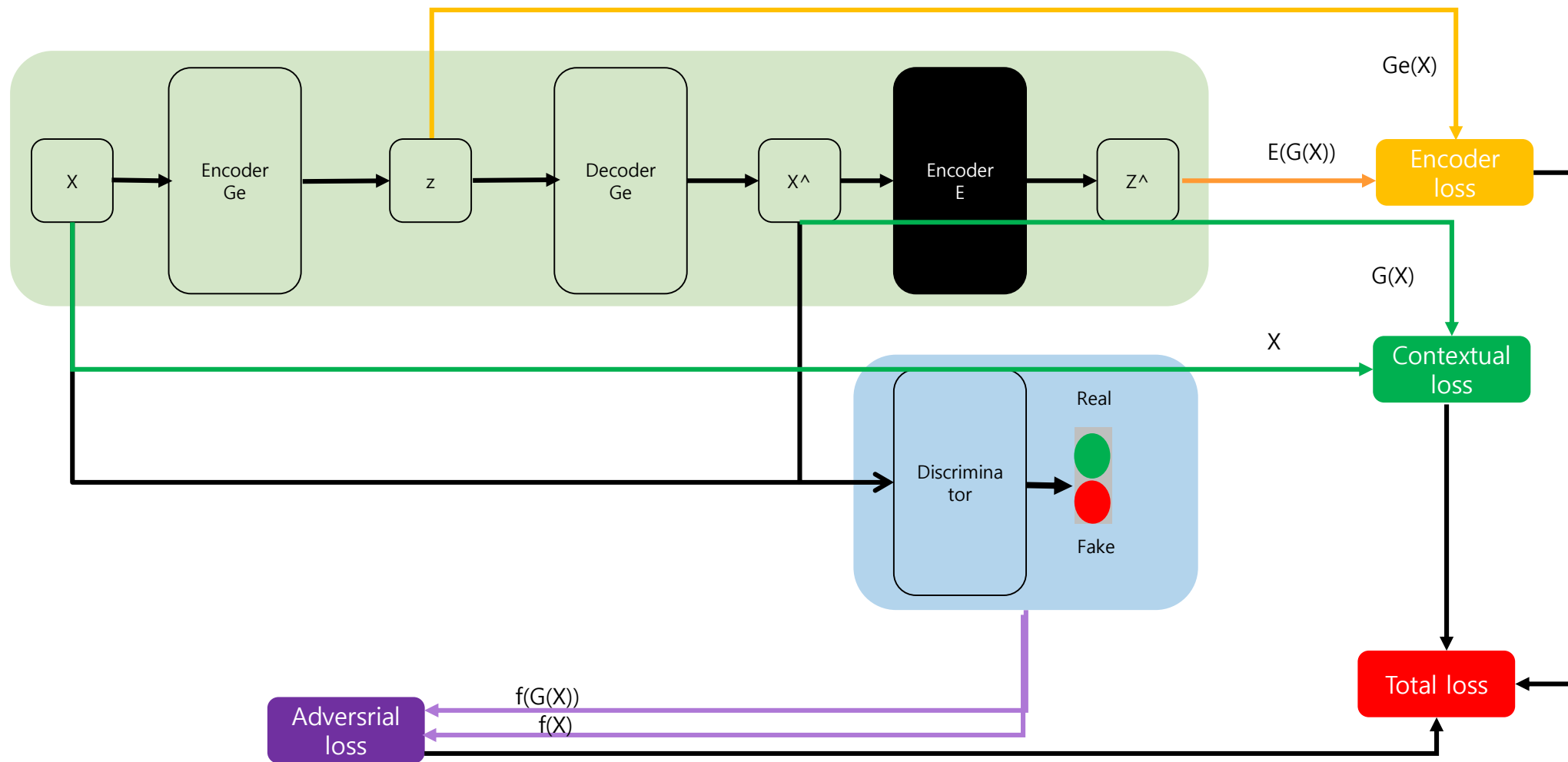
$$\mathcal{A}(\mathbf{x}) = \|G_E(\mathbf{x}) - E(G(\mathbf{x}))\|_2 .$$

$$s'_i = \frac{s_i + \min(S)}{\max(S) - \min(S)} .$$

\*Anomaly Score: GANomaly의 Encoder Loss를 바탕으로 계산

\*0~1 사이의 Anomaly Score Scaling 진행

### 3-5. GANomaly



출처: GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training

## 3-6. Skip-GANomaly

### <GANomaly 구조>

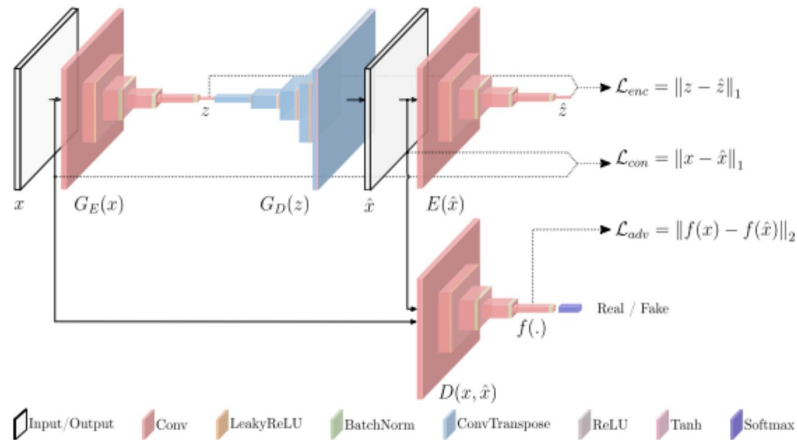
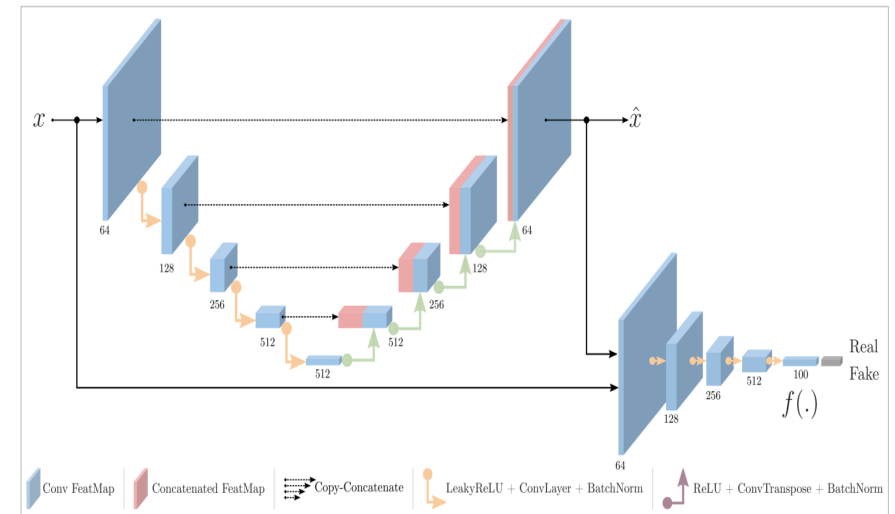


Figure 3. GANomaly architecture and loss functions from (Akçay et al., 2018).

### <Skip-GANomaly 구조>

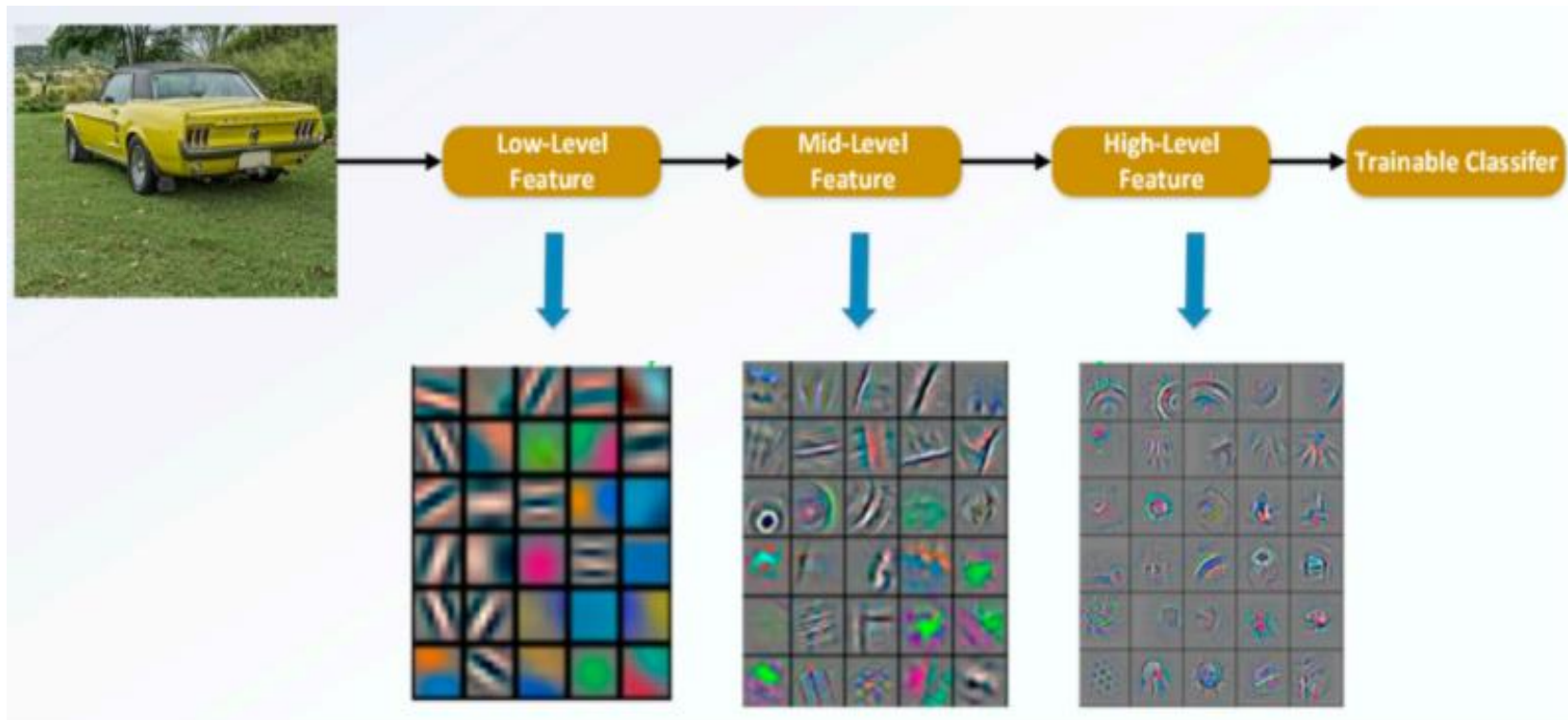


### \*Skip-GANomaly

- GANomaly 모델에서 **Skip-connection**을 추가한 모델
- Skip-connection을 통해 계층들간에 정보를 직접적으로 전달하여 이전 Layers 정보를 더 잘 전달할 수 있음

## 3-6. Skip-GANomaly

### <CNN의 학습 구조>

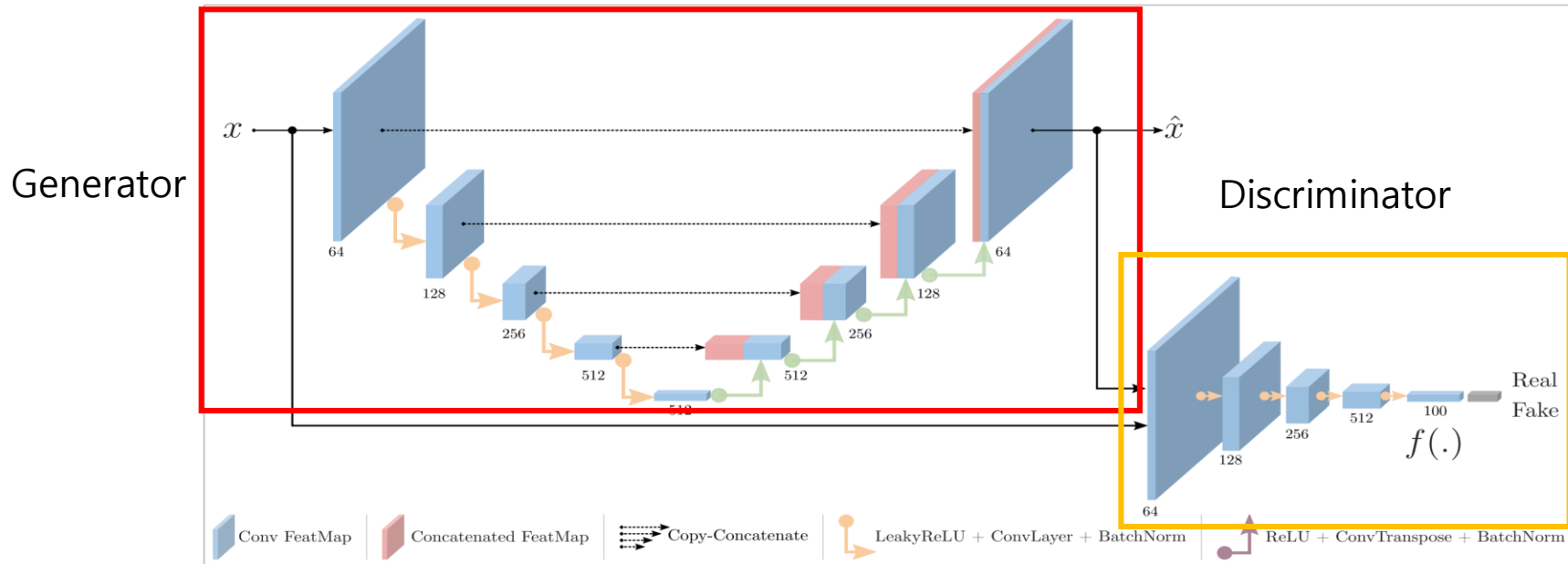


### \*Skip-Connection 사용이유

- CNN 학습 과정에서 망이 깊어지면서 이전 Feature의 영향도가 떨어지기 때문에 학습이 잘 되지않는 문제 발생
- 각 층에 직접 정보를 전달하는 Layer를 추가하는 방식의 Skip- Connection을 통해 문제해결

## 3-6. Skip-GANomaly

<Skip-GANomaly 구조>



### \*Skip-GANomaly

- Generator 부분에는 GANomaly와 같이 AutoEncoder를 사용하며, Encoder와 Decoder는 같은 층으로 구성
- Encoder의 정보를 Decoder에 직접 전달하는 Skip Connection을 통해 모델을 학습

## 1. Anomaly Detection 개념 및 적용사례

## 2. Anomaly Detection 종류

- 학습유형에 따른 Anomaly Detection
- 비정상 데이터 종류에 따른 Anomaly Detection

## 3. Anomaly Detection 알고리즘 소개

- 딥러닝 생성모델(AutoEncoder, GAN)
- Anogan
- Fast-Anogan
- GANomaly
- Skip-GANomaly

## 4. Image Anomaly Detection 실습

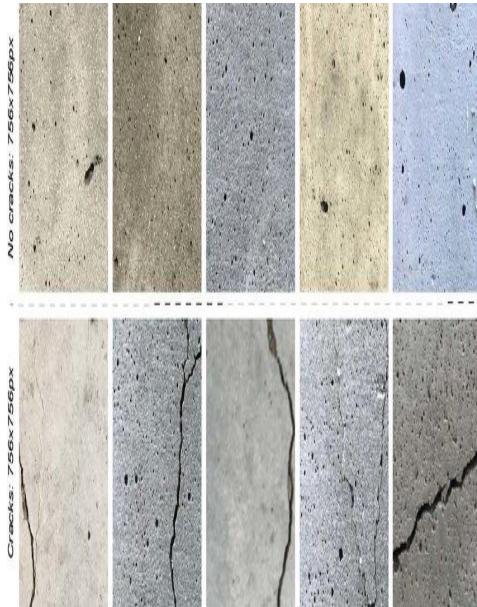
- 건축 분야: 건물의 균열 데이터
- 의료 분야: 코로나 환자 흉부 X-ray 데이터
- 제조 분야: 물품의 흠집, 파손데이터

## 5. 결론 및 활용방안

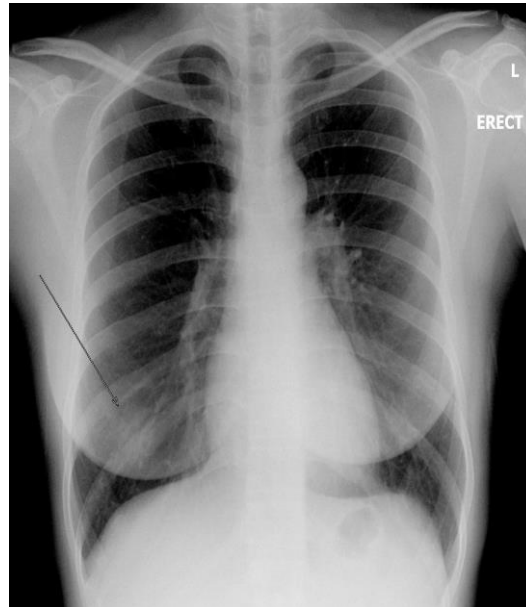


## 4-1. 실습 분야

<건축 분야: 벽의 균열 탐지>



<의료 분야: 코로나 환자 탐지 >



<제조 분야: 불량품 탐지>



\*Kaggle에서 수집한 존재하는 각 분야의 실제 데이터 활용

\*Anomaly Detection 모델의 평가를 위해 정상과 비정상 데이터의 Label이 존재하는 평가 데이터 활용

데이터 출처: <https://www.kaggle.com/anjana22/concrete-crack-detection>

## 4-1. 실험환경

### \*실험 환경(사내 GPU 서버)

#### \*서버 사양

- GPU: GTX 1080 11GB(2대)
- Nvidia- Driver: 387.26
- Cuda Version: 9.0


#### \*사용한 딥러닝 프레임 워크

- F-anogan: Keras, pytorch
- Ganomaly: Keras, pytorch

#### \*주요 패키지 버전

- python: 3.6 ver
- tensorflow-gpu: 1.14 ver
- keras: 2.3.1 ver
- Cudnn: 7.1.2 ver

### <사용한 딥러닝 프레임워크의 특징>

 Keras	 PYTORCH
<ul style="list-style-type: none"><li>• Tensorflow Backend를 이용 (1.4버전 이후 통합됨)</li><li>• 모델을 만들고, 실행하는 이해하기 쉬운 구조로 이루어져 있음</li><li>• Resnet, VGG net등 Keras 자체내에서 제공하는 모델을 Input shape 만 맞추면 학습할 수 있음</li><li>• 모델 학습시 사용하는 GPU의 Memory를 모두 점유함(Tensorflow 2.0 이후 문제 해소)</li></ul>	<ul style="list-style-type: none"><li>• 2-3년 전부터 각광 받고 있는 딥러닝 프레임 워크</li><li>• Data loader를 만들어서 batch size 만큼 데이터를 추출하고 이를 모델에 학습하는 것이 특징</li><li>• 필요한 만큼만 GPU를 사용함</li><li>• 직관적이지 않은 코드로 이해하기 어려움</li></ul>

## 4-2. Image Anomaly Detection 실습

---

- 실험방법

- 1)Image Scale: Gray image, Full color image
- 2)Image Size: 64, 128
- 4)Batch size: 16, 32, 64
- 5)Learning Late: 0.001, 0.0005
- 6)Activation: Leaky Relu, Batch Normalization 사용
- 7)Image Augmentation 기법 적용

- 평가지표

- \* 정량적 지표

- Min-max Scaling을 적용한 0-1사이 값의 Anomaly Score 사용
  - 비정상 데이터는 Anomaly Score가 높다는 전제하에 Anomaly Score를 Anomaly에 대한 확률 값으로 가정하여 **AUC** 및 **AUPR** 계산
  - 비정상 데이터와 정상데이터의 Anomaly Score Boxplot 및 분포 확인

- \* 정성적 지표

- 비정상 이미지와 실제 이미지의 다른 부분을 육안으로 확인

- 실험데이터

\*캐글에서 존재하는 각 분야의 실제 데이터를 활용

- 1)건축 분야

- 건물 콘크리트 벽의 균열 Detection

- 2)의료 분야

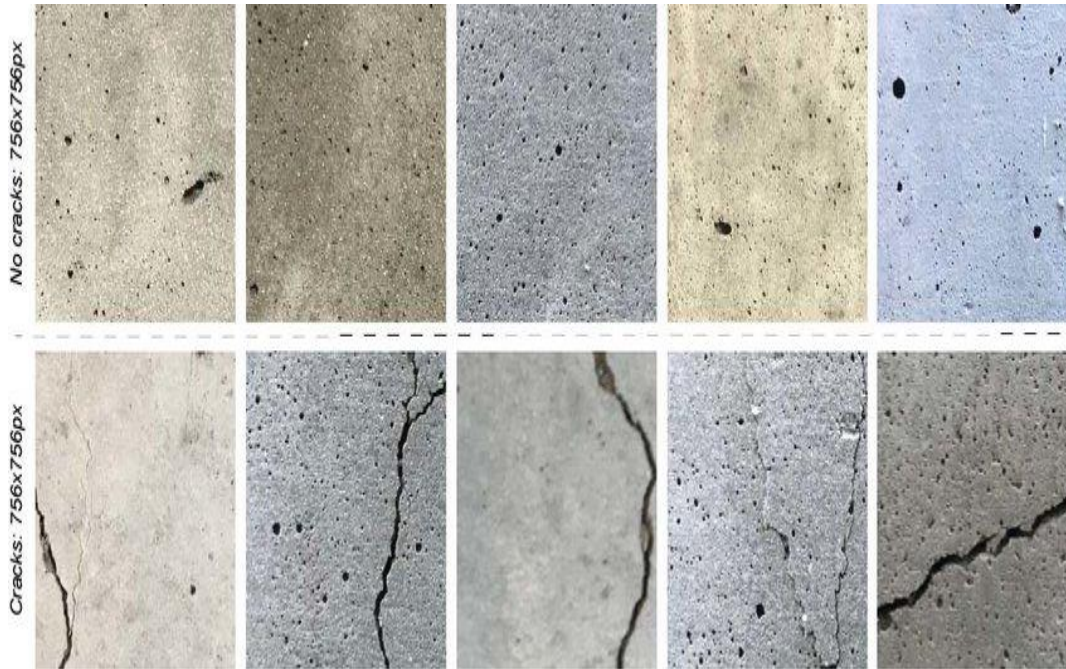
- 폐 손상 여부를 통해 코로나 환자 Detection

- 3)제조 분야

- 제조 물품 중 불량품 Detection

## 4-5. 건축 분야 실습

### <건물 균열 데이터>



#### \*콘크리트 벽의 균열 데이터

- 건물의 균열 탐지 모델 구축

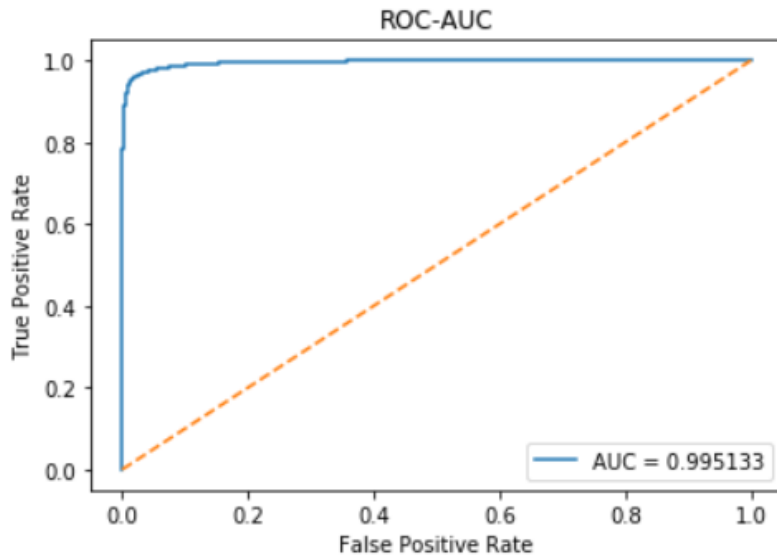
- Train Data
  - 정상데이터: 15,000 images
- Test Data
  - 정상데이터: 5,000 images
  - 비정상 데이터: 5,000 images

데이터 출처: <https://www.kaggle.com/anjana22/concrete-crack-detection>

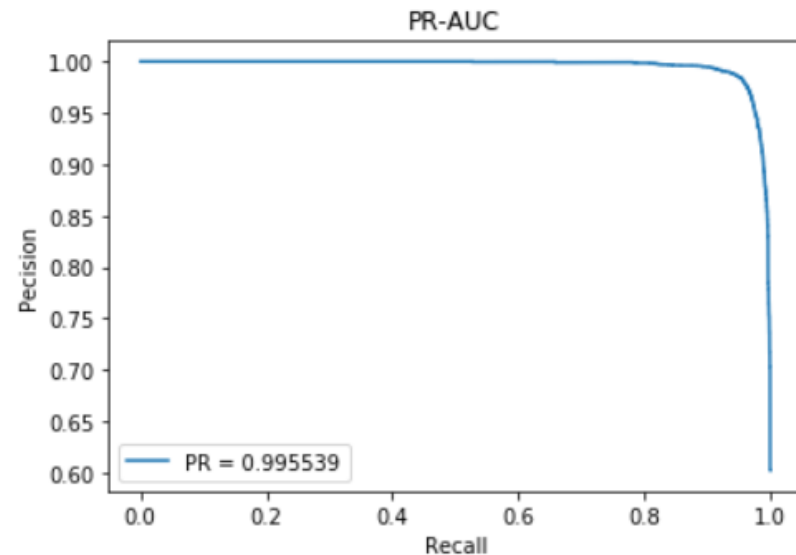
## 4-5. 건축 분야 학습결과

- Ganomaly와 F-Anogan의 모델 최적화시 **F-Anogan**모델에서 AUC 및 AUPR Score값이 0.995로 가장 높았음  
(Model Option: Full color, Image size 64, epoch 500, learning late 0.0005)
- 총 학습시간: 3시간 45분 (Generator 학습시간: 1시간 45분, Encoder 학습시간: 2시간)

<AUC Score>



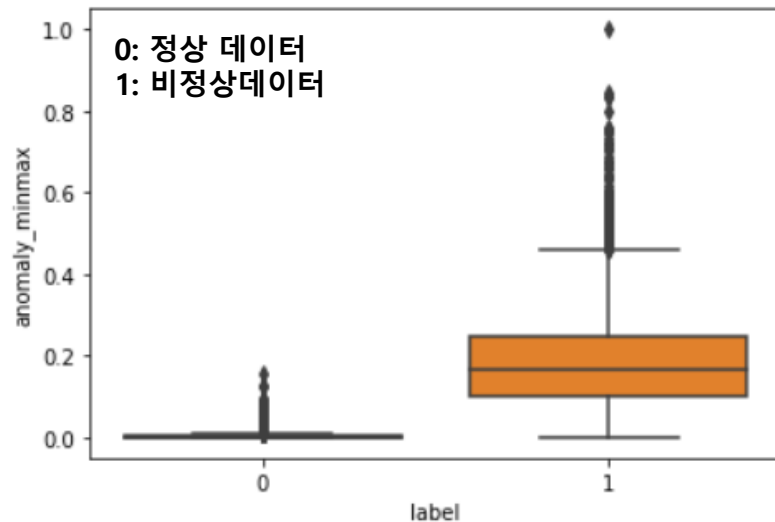
<AUPR Score>



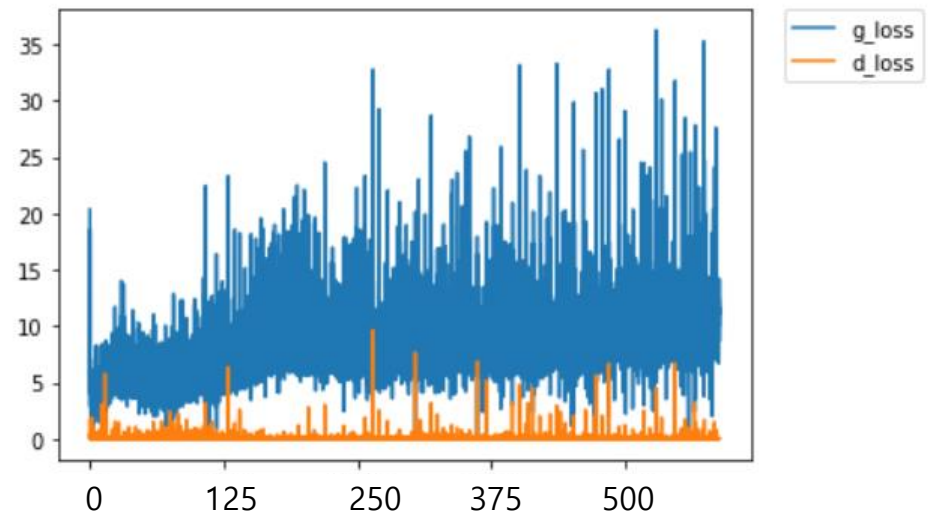
## 4-5. 건축 분야 학습결과

- 비정상 데이터의 Anomaly Score 평균이 정상 데이터 Anomaly Score의 평균보다 높으며, 정상 데이터의 편차는 적은 반면, 비정상 데이터의 편차는 큼
- 학습결과는 좋으나, Generator loss가 Discriminator loss보다 높으며, Generator Model의 loss가 Epoch 마다 불안정함

<Boxplot 비교>

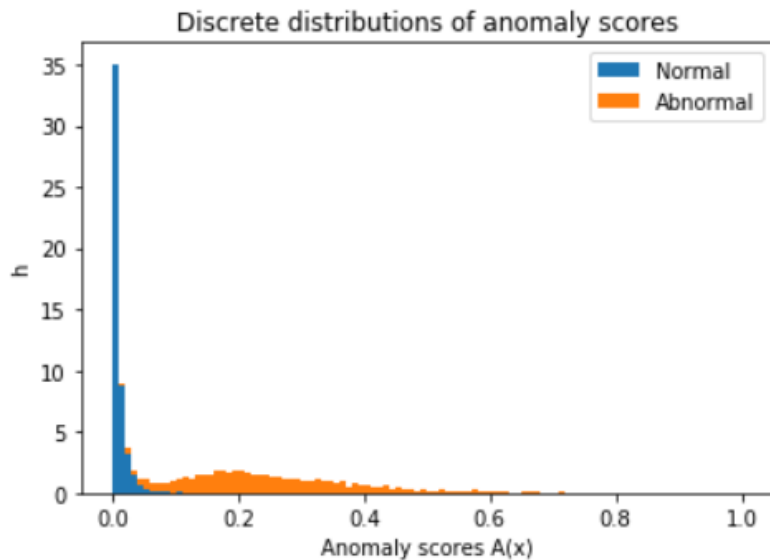


<Epoch에 따른 Loss 값>

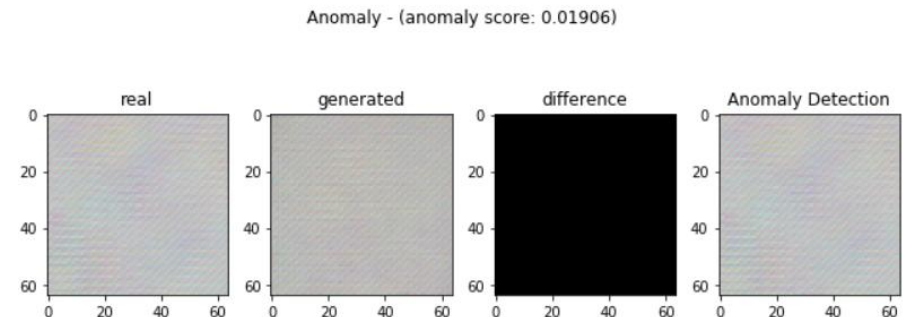


## 4-5. 건축 분야 학습결과

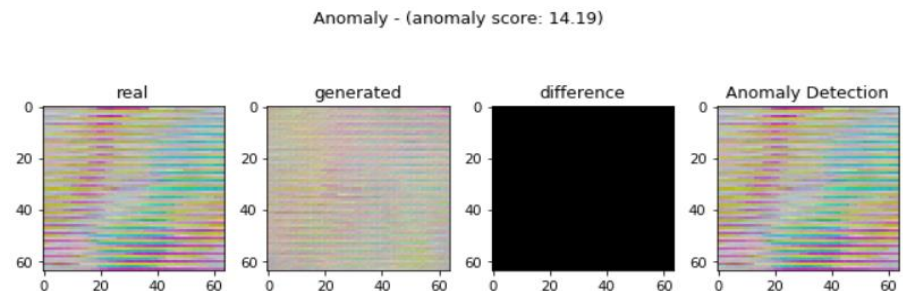
### <정상데이터와 비정상 데이터의 분포>



### <정상 데이터 이미지 비교>



### <비정상 데이터 이미지 비교>



- 비정상 데이터의 Anomaly Score 분포가 정상 데이터의 Anomaly Score에 비해 오른쪽으로 쏠려 있으며, 정상 데이터의 Anomaly Score 값들이 아주 낮은 것을 확인할 수 있음
- 정상이미지와 생성한 이미지의 Anomaly Score는 0.01906로 낮은 반면, 비정상 데이터와 생성한 이미지의 Anomaly Score는 14.19로 값이 큰 것을 확인할 수 있음



## 4-6. 의료 분야

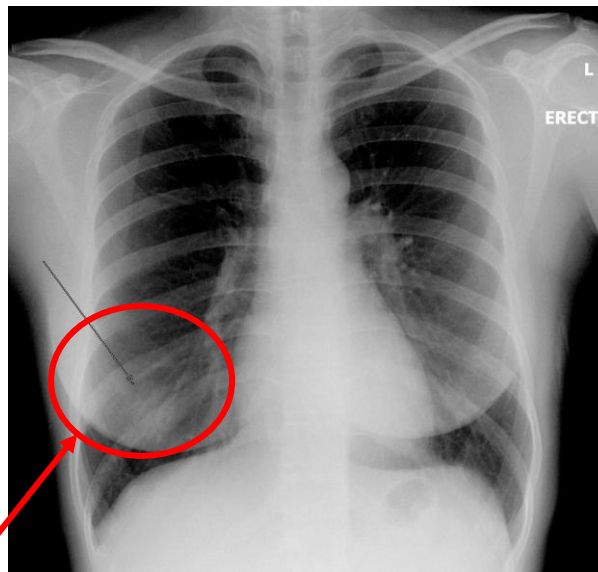
### \*코로나 환자와 정상인의 흉부 X-ray 데이터

- 코로나 환자의 경우 폐 손상으로 X-ray 상에 흐리게 보임

<정상 데이터의 흉부>



<비정상 데이터의 흉부>



폐 손상부위

- Train Data
  - 정상데이터: 70 images
- Test Data
  - 정상데이터: 20 images
  - 비정상 데이터: 26 images

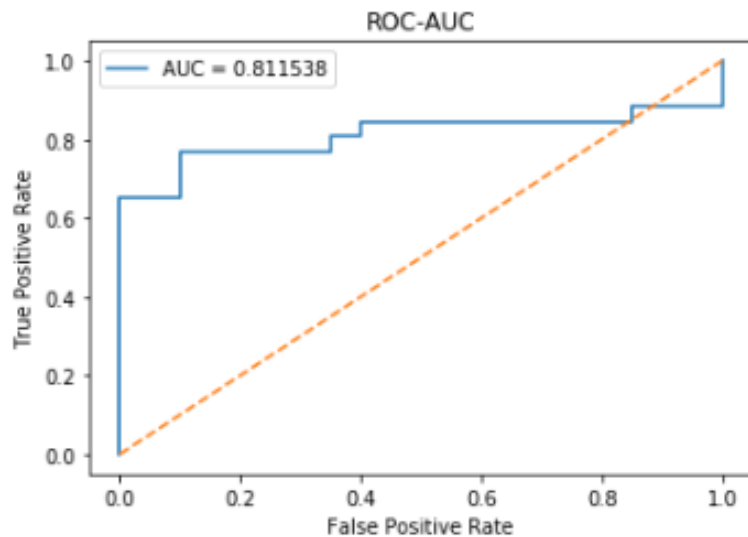
출처: <https://www.kaggle.com/pranavraikokte/covid19-image-dataset>



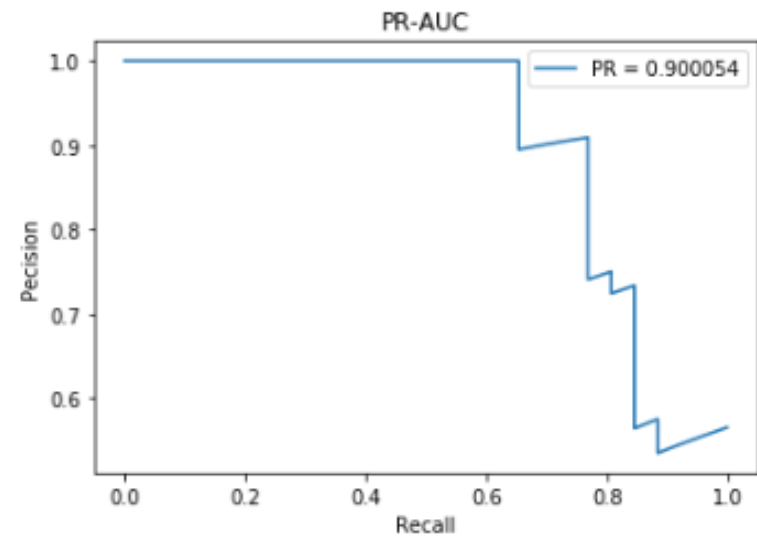
## 4-6. 의료 분야 학습결과

- Ganomaly와 F-anogan의 모델을 최적화시 **GANomaly Model**에서 AUC 및 AUPR Score가 가장 좋았음  
(Model Option: Gray color, Image size 128, epoch 100, learning late 0.001)
- AUC: 0.811, AUPR: 0.90
- 총 학습시간: 12분

<AUC Score>



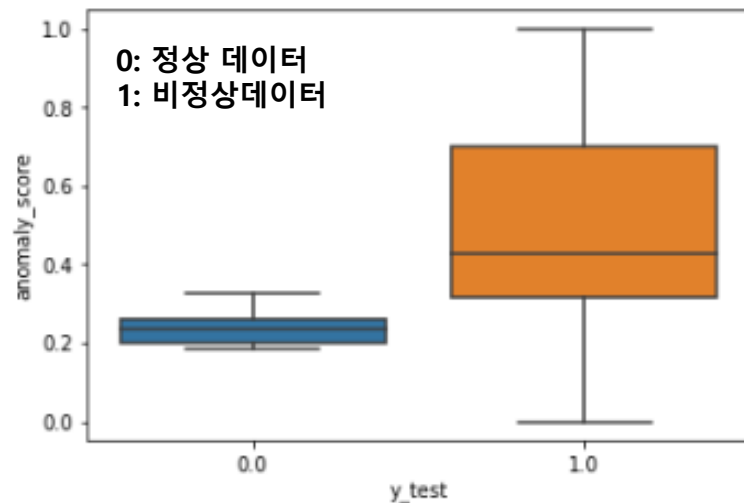
<AUPR Score>



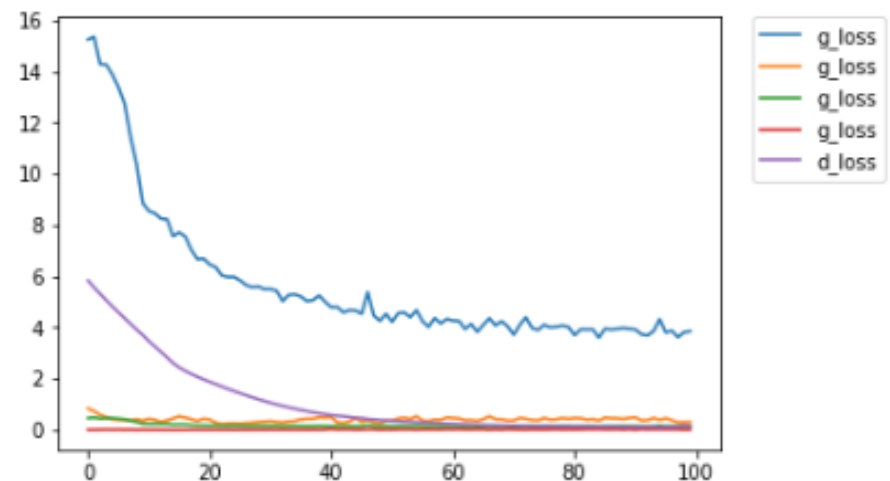
## 4-6. 의료 분야 학습결과

- 비정상 데이터의 Anomaly Score 평균이 정상 데이터 Anomaly Score의 평균보다 높으며, 정상 데이터의 편차는 적은 반면, 비정상 데이터의 편차는 큼
- Generator loss가 Discriminator loss보다 높으며, Epoch이 50일 때 Loss가 안정적으로 수렴하기 시작함  
(파란색 선: Generator total loss, 보라색 선: Discriminator total loss)

<Boxplot 비교>

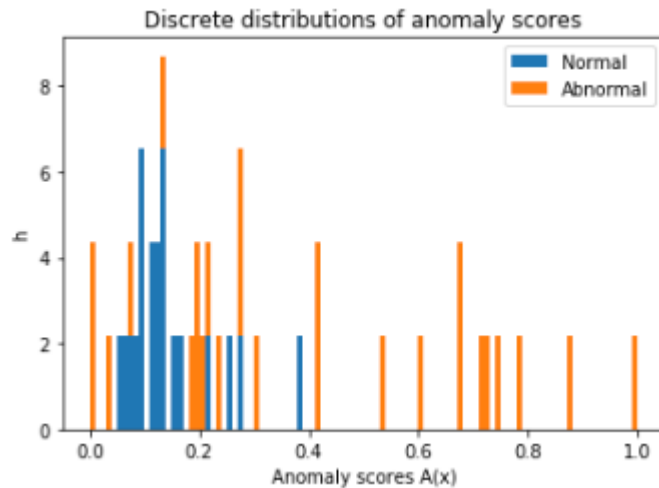


<Epoch에 따른 Loss 값>



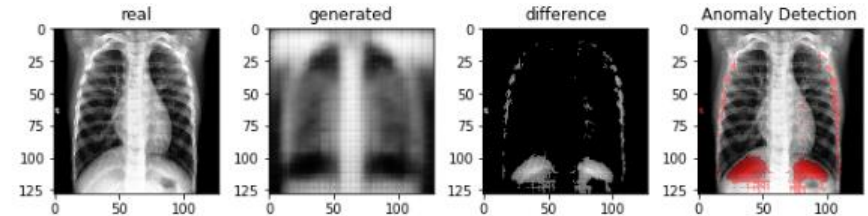
## 4-6. 의료 분야 학습결과

### <정상데이터와 비정상 데이터의 분포>



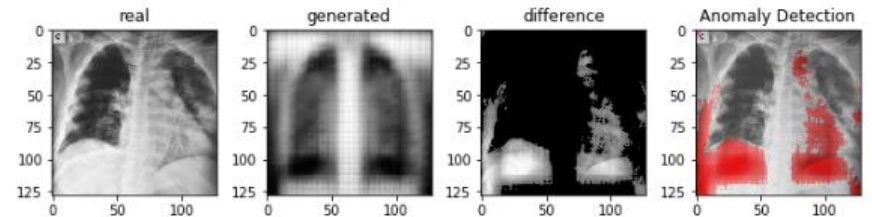
### <정상 데이터 이미지 비교>

Anomaly - (anomaly score: 0.3899)



### <비정상 데이터 이미지 비교>

Anomaly - (anomaly score: 0.8732)



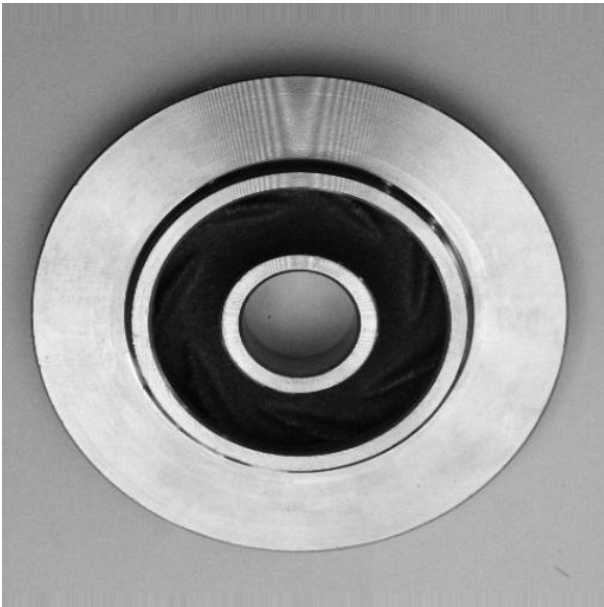
- Anomaly Score 분포가 정상 데이터의 Anomaly Score에 비해 오른쪽으로 쏠려 있으며, 정상 데이터의 Anomaly Score 값들이 낮은 것을 확인할 수 있음
- 정상이미지와 생성한 이미지의 Anomaly Score는 0.3899로 낮은 반면, 비정상 데이터와 생성한 이미지의 Anomaly Score는 0.8732로 높은 것을 확인할 수 있음
- Detection 판단 위치(빨간색 부분)가 애매하며, 결과 해석을 위해 의학적 지식이 필요함(폐 손상 위치등)

## 4-7. 제조 분야

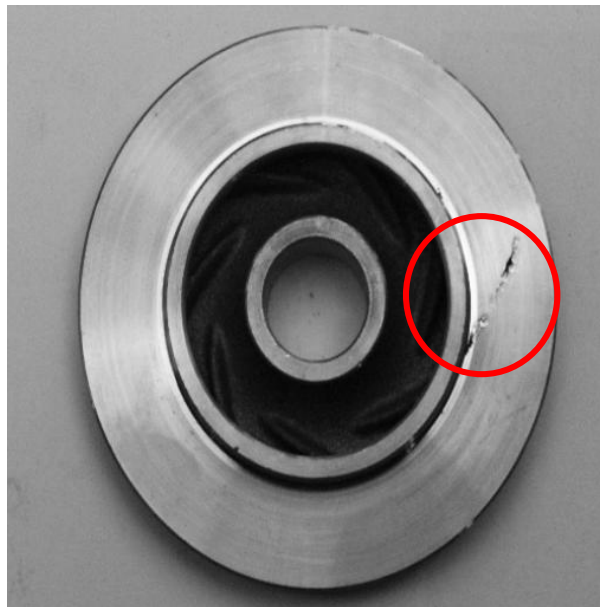
### \*제조 물품의 불량품 탐지데이터

-불량품에는 흠집, 파손 등의 손상이 가해져 있음

<정상 물품>



<파손 물품>



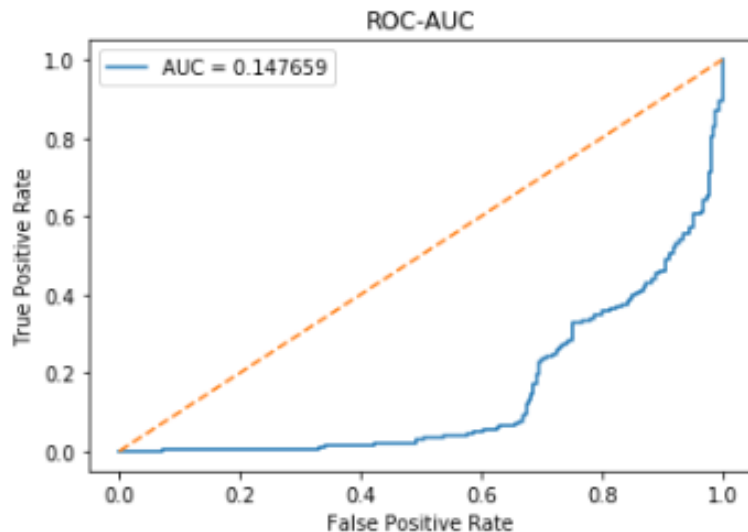
- Train Data
  - 정상데이터: 2,872 images
- Test Data
  - 정상데이터: 262 images
  - 비정상 데이터: 453 images

출처: <https://www.kaggle.com/pranavraikokte/covid19-image-dataset>

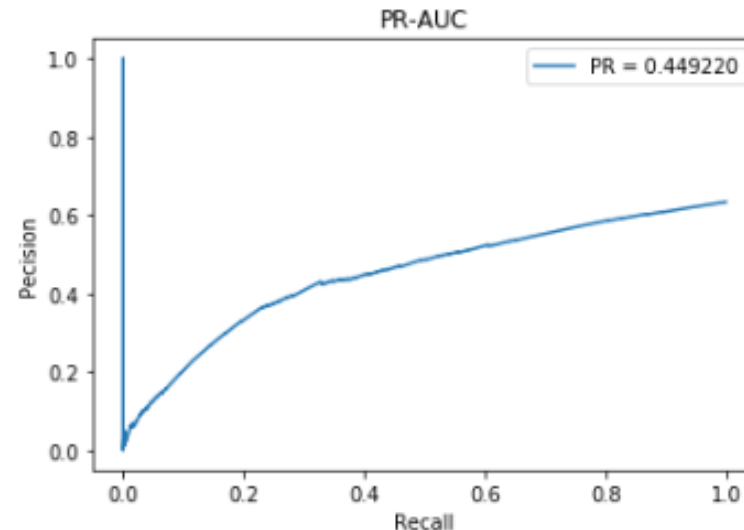
## 4-7. 제조 분야 학습결과

- Ganomaly와 F-anogan 모델 모두 학습이 제대로 되지 않으며, 학습이 잘되지 않은 GANomaly 모델의 예시결과  
(Model Option: gray scale, 128 image size, epoch 100, learning late 0.001)
- 총 학습시간: 12분
- AUC와 AUPR모두 모든 값을 정답으로 했을 때의 Score값인 0.5보다 낮은 것을 확인할 수 있음

<AUC Score>



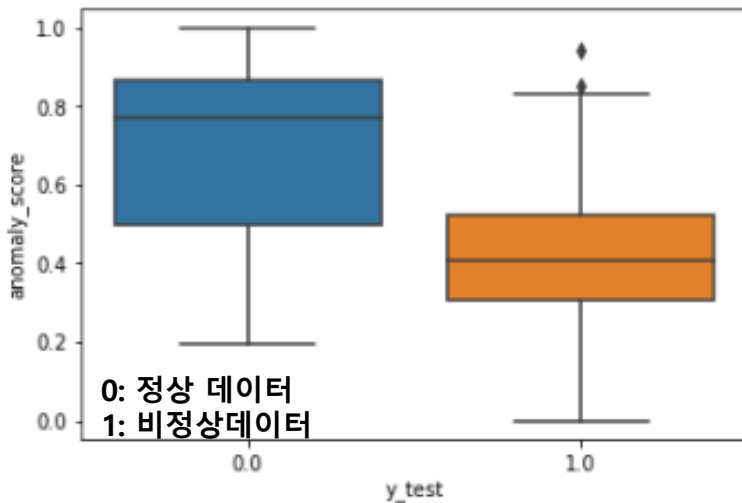
<AUPR Score>



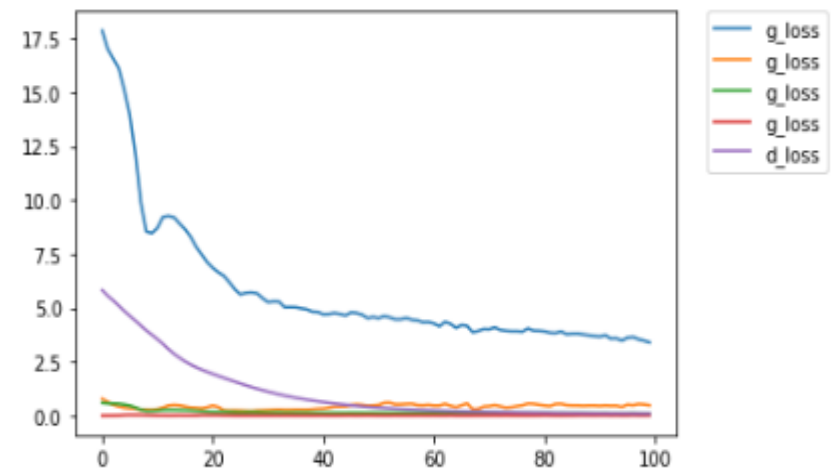
## 4-7. 제조 분야 학습결과

- 정상데이터의 Anomaly Score 평균이 비정상 데이터의 Anomaly Score 평균 보다 높으며, 각 데이터들의 Anomaly Score 편차가 큼
- Generator loss가 Epoch마다 떨어지며 안정적으로 학습을 진행하고 있으나, 학습결과는 좋지 않음  
(파란색 선: Generator total loss, 보라색 선: Discriminator total loss)

<Boxplot 비교>

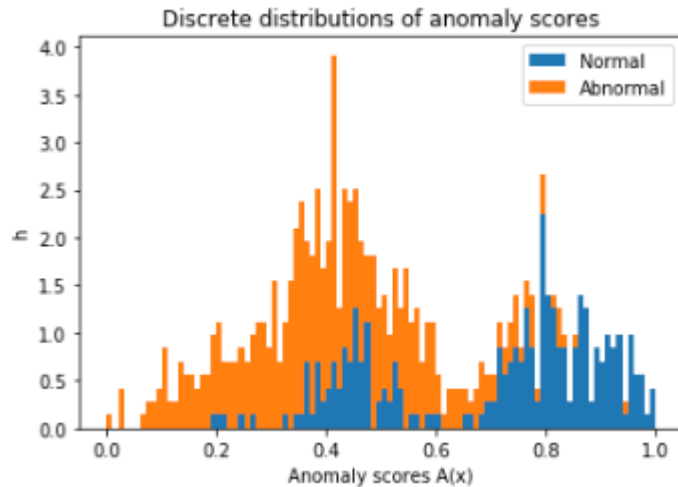


<Epoch에 따른 Loss 값>



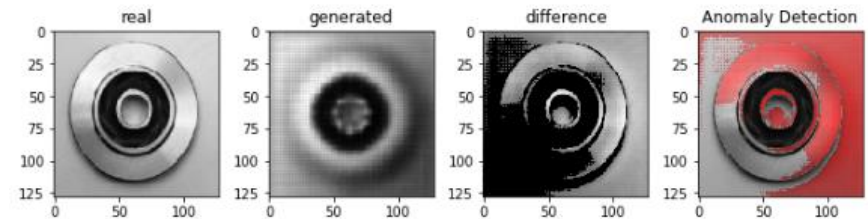
## 4-7. 제조 분야 학습결과

### <정상데이터와 비정상 데이터의 분포>



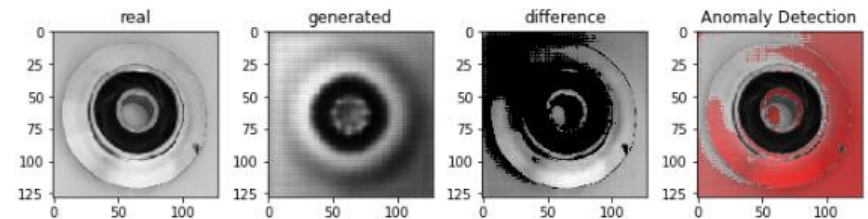
### <정상 데이터 이미지 비교>

Anomaly - (anomaly score: 0.806)



### <비정상 데이터 이미지 비교>

Anomaly - (anomaly score: 0.8242)



- 정상 데이터의 Anomaly Score 분포가 비정상 데이터의 Anomaly Score에 비해 오른쪽으로 쏠려 있으며, 정상 데이터의 Anomaly Score 값들이 전반적으로 높음
- 정상이미지와 생성한 이미지의 Anomaly Score는 0.806이고 비정상 데이터와 생성한 이미지의 Anomaly Score는 0.8242로 차이가 없음
- Detection 위치(빨간색)에 물품의 파손부위가 들어가지만 Detection 위치가 애매함

## 1. Anomaly Detection 개념 및 적용사례

## 2. Anomaly Detection 종류

- 학습유형에 따른 Anomaly Detection
- 비정상 데이터 종류에 따른 Anomaly Detection

## 3. Anomaly Detection 알고리즘 소개

- 딥러닝 생성모델(AutoEncoder, GAN)
- Anogan
- Fast-Anogan
- GANomaly
- Skip-GANomaly

## 4. Image Anomaly Detection 실습

- 건축 분야: 건물의 균열 데이터
- 의료 분야: 코로나 환자 흉부 X-ray 데이터
- 제조 분야: 물품의 흠집, 파손데이터

## 5. 결론 및 활용방안



## 5. 활용방안 및 결론

---

### • 결론 및 시사점

- 1) Ganomaly 모델이 F-Anogan 모델의 학습시간보다 전반적으로 짧음
- 2) Image size(64, 128) 에 따라 모델의 구동 소요시간의 차이가 큼
- 3) **Anomaly Score의 절대적인 기준이 없으며, 모델마다 Anomaly Data로 판단하는 Score의 기준이 다름**
- 4) 정상데이터와 비정상 데이터가 뚜렷하게 구분 될 수 있는 데이터를 활용해야 학습 가능
  - 제조 분야 Anomaly Detection 모델의 실패 원인
- 5) 정상데이터의 이미지에 일관성이 있어야 학습이 가능함  
(Ex. 의료분야의 X-ray 데이터)
- 6) 의료분야에서 적용 시 결과에 대한 전문가의 해석이 필요함

### • 활용방안

#### \* 이상 탐지 프로젝트( Ex. BMS, KT통신구 화재 예측 등)

- GAN은 Generator와 Discriminator에 여러 가지 모델을 사용할 수 있기 때문에, 모델의 아키텍처를 바꾸면 **정형 데이터 이용한** 이상치 탐지 분야에서 사용할 수 있음(2D CNN → 1D CNN)
- BMS 프로젝트와 통신구 화재 예측 프로젝트의 데이터는 단일 변수이지만, 새로운 변수를 추가하면 모델의 적용가능성이 있음
- 단, 비정상 데이터의 Anomaly Score 기준을 정하는 것이 필요하며 생성한 이미지와 실제 이미지를 비교하는 정성적인 평가 불가능  
(기존 분석모델과 비교를 통한 검증 필요)
- 딥러닝을 통한 이상 탐지 예측모델 중 하나의 안으로 제시 가능

---

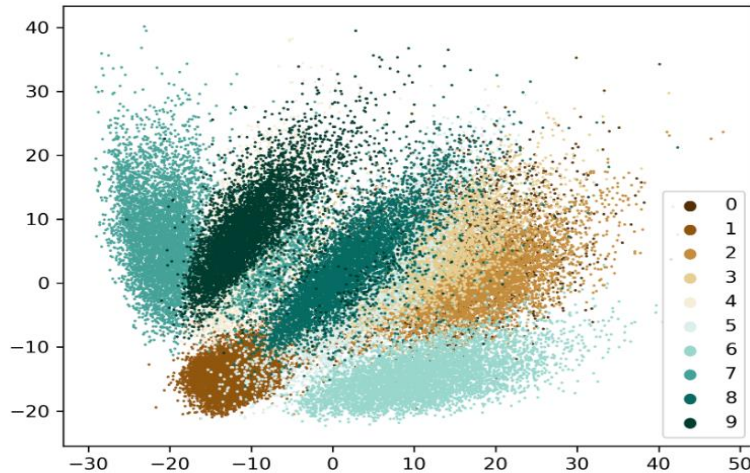
# 감사합니다

---

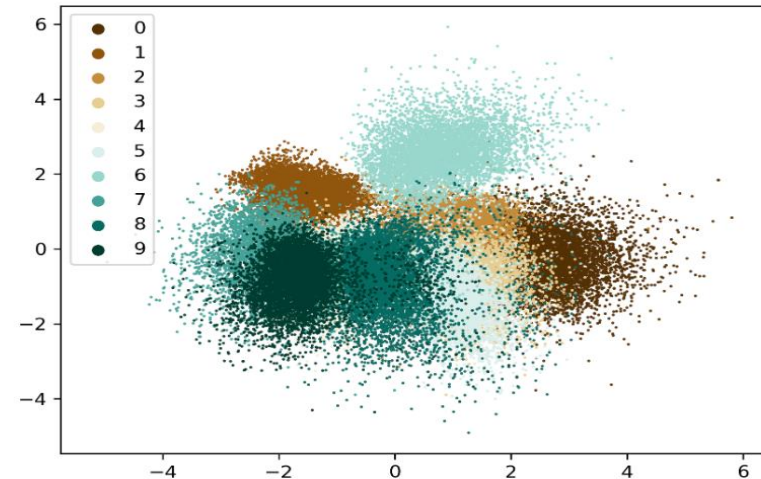
# Appendix

## \* AutoEncoder 와 Variational AutoEncoder 비교

<AutoEncoder Latent space>



<Variational AutoEncoder Latent space>



### \*Mnist로 학습한 AutoEncoder 와 Variational AutoEncoder의 Latent space 비교 예시

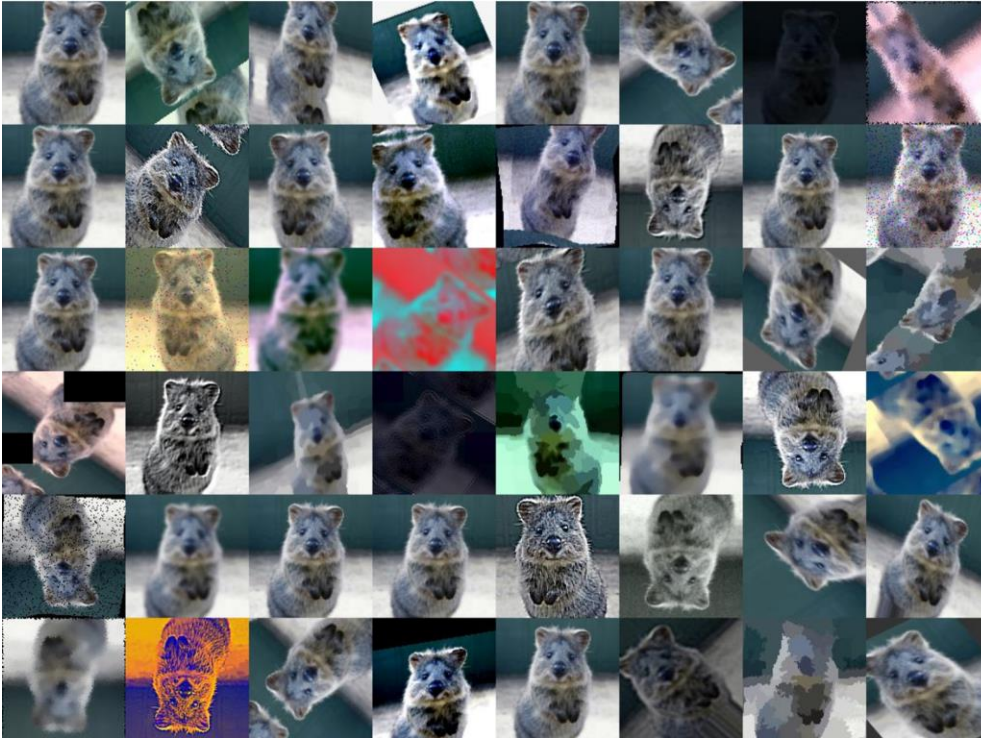
- AutoEncoder 보다 Variational AutoEncoder의 Latent Space의 숫자 값들이 더 밀집되어 있으며, 구분이 뚜렷함

### \*AutoEncoder의 단점

- Reconstruction Error 사용: 실제이미지와 생성이미지의 차이가 줄어들도록 학습하기 때문에 생성한 이미지가 실제이미지와 흡사하며 생성한 이미지의 다양성이 부족함

# \* Image Augmentation

## <Image Augmentation>



### \* Image Augmentation이란?

- 학습데이터 이미지의 각도, 색상등을 여러 방법으로 조절하여 딥러닝 모델에 훈련
- 모델이 1가지 이미지에 대해 여러 유형을 학습할 수 있어 정확도 상승
- Pytorch와 Keras 자체 Function을 통해 구현이 가능함
- 각 분야에서 실습한 데이터에 Image Augmentation 적용