BE530 – Medical Deep Learning

- Generative Adversarial Networks (GANs) -

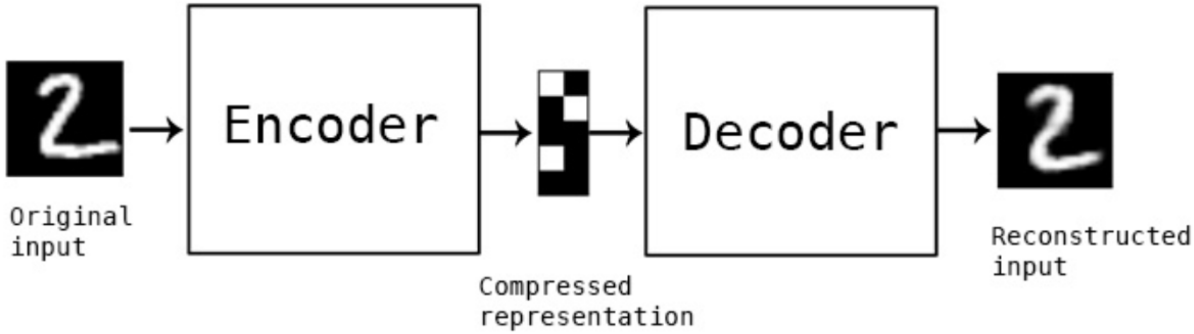
Byoung-Dai Lee

Division of AI Computer Science and Engineering

Kyonggi University

KYONGGI UNIVERSITY

AutoEncoder

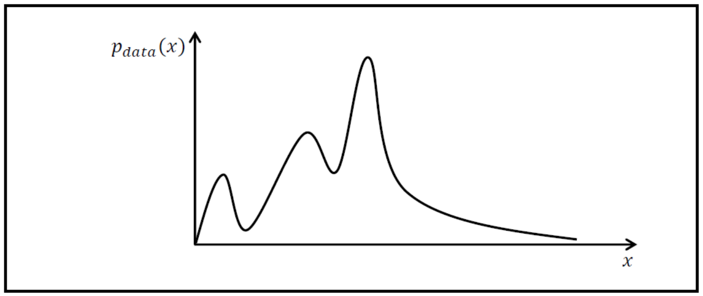


- Encoding 과정을 통해 실제 데이터보다 차원을 축소하여 compressed representation을 생성하고 다시 Decoder를 통해 차원을 증가시켜 원본 데이터를 복원함
  - Compressed representation을 복원 이미지 "2"를 만들기 위한 latent variable(잠재 변수)라고 볼 수 있음
- 즉 어떤 random noise가 있더라도 decoder를 잘 학습시킬 수 있으다면 원하는 분포를 갖는 데이터를 생성할 수 있음
- Generator란 random input z를 원하는 분포를 갖는 데이터로 생성하는 것으로 볼 수 있음

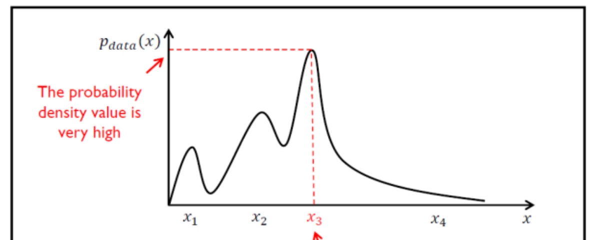
Generative Model

Probability density function

There is a $p_{data}(x)$ that represents the distribution of actual images.

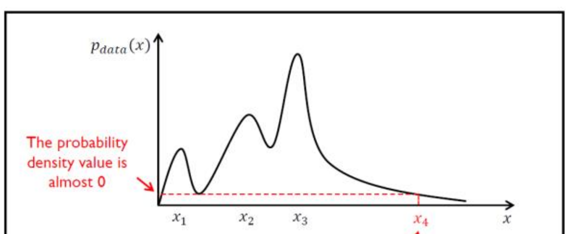


Our dataset may contain very many images of women with blonde hair.



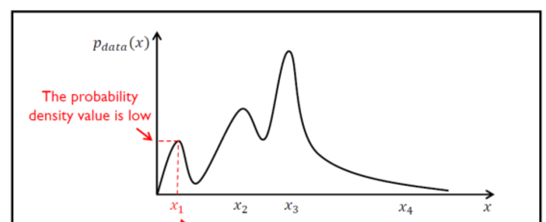
x_3 is a 64x64x3 high dimensional vector representing a woman with blonde hair.

Our dataset may not contain these strange images.



x_4 is an 64x64x3 high dimensional vector representing very strange images.

Let's take an example with human face image dataset.
Our dataset may contain few images of men with glasses.



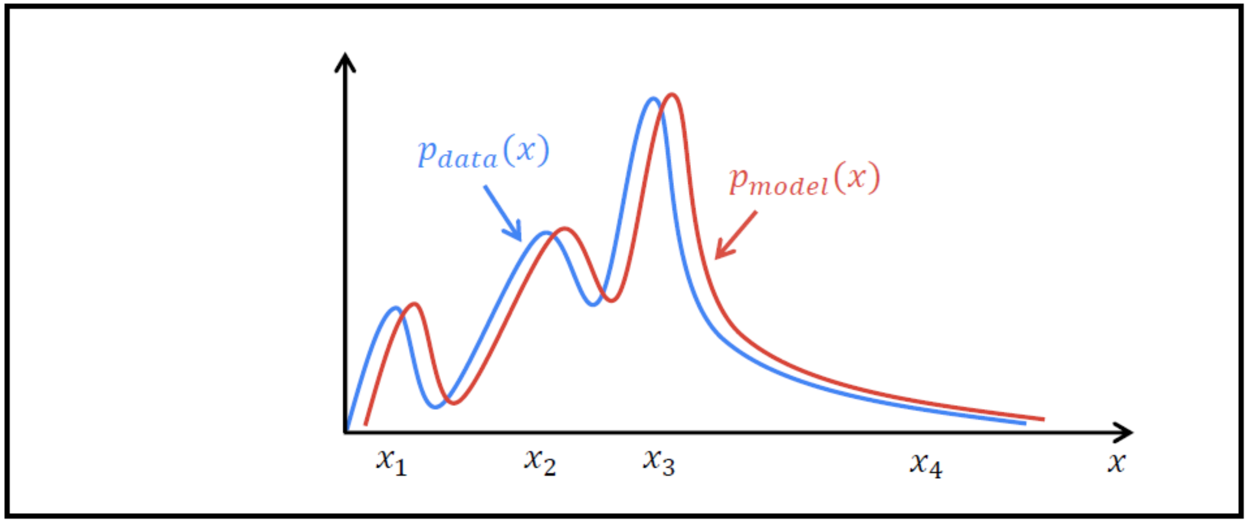
x_1 is a 64x64x3 high dimensional vector representing a man with glasses.

Generative Model (cont.)

Distribution of images generated by the model

The goal of the generative model is to find a $p_{model}(x)$ that approximates $p_{data}(x)$ well.

Distribution of actual images



Generative Model (cont.)

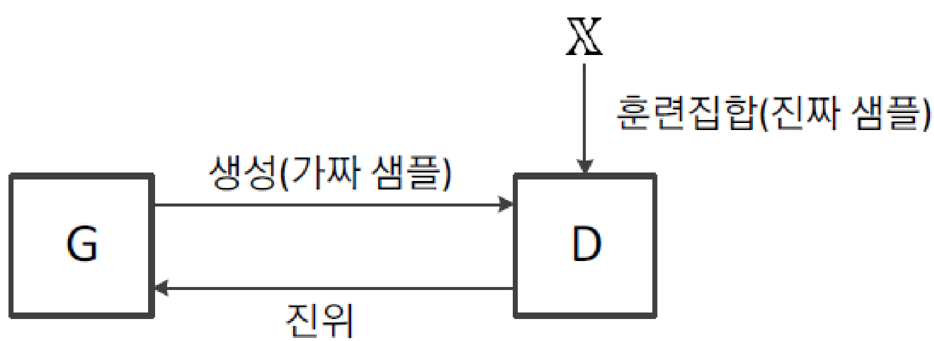
■ 분별 모델과 생성 모델의 비교

모델	학습 단계가 할 일	예측 단계가 할 일	지도 여부
분별 모델	$P(y\|\mathbf{x})$추정	$f: \mathbf{x} \mapsto y$	지도 학습
생성 모델	$P(\mathbf{x})$ 또는 $P(\mathbf{x}\|y)$, $P(\mathbf{x}, y)$ 추정	f: 씨앗$\mapsto$ $\mathbf{x}$ 또는 f: 씨앗, $y \mapsto \mathbf{x}$, f: 씨앗$\mapsto$ $\mathbf{x}, y$	비지도 학습

GAN

The concept of GAN

- 생성기 G와 분별기 D의 대립 구도
  - G는 가짜 샘플 생성(위조지폐범)
  - D는 가짜와 진짜를 구별(경찰)
- GAN의 목표는 위조지폐범의 승리(G가 만들어내는 샘플을 D가 구별하지 못하는 수준까지 학습)

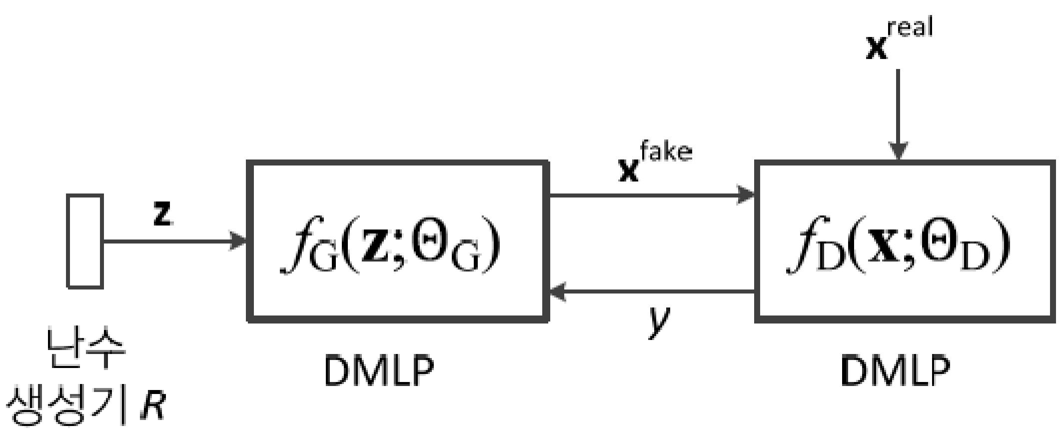


KYONGGI UNIVERSITY

GAN (cont.)

■ 최초 GAN

- G와 D를 DMLP로 구현
  - G는 $f_G(z;\Theta_G)$, D는 $f_D(z;\Theta_D)$로 표기 (Θ_G와 Θ_D는 매개변수)
  - f_G는 난수 발생기로 만든 벡터 z를 입력으로 받아 가짜 영상을 출력
  - f_D는 영상을 입력으로 받아 진짜(1) 또는 가짜(0)를 출력



Objective Function of GAN

Sample x from real data distribution

Sample latent code z from Gaussian distribution

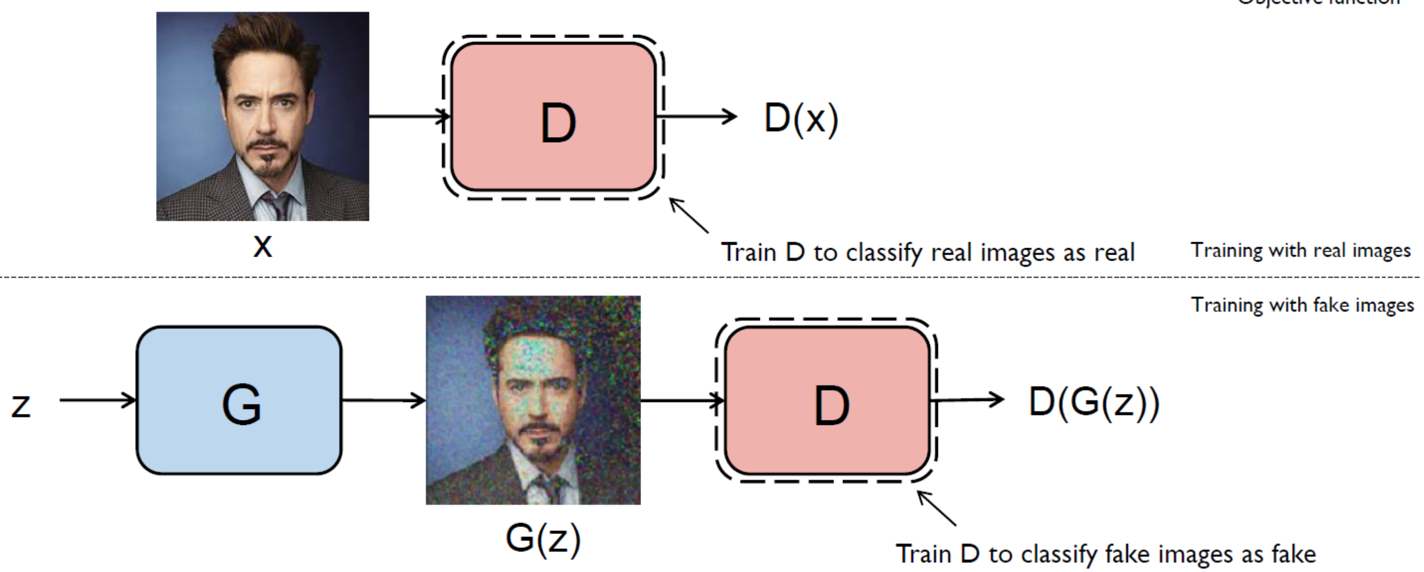
$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

D should maximize $V(D,G)$

Maximum when $D(x) = 1$

Maximum when $D(G(z)) = 0$

Objective function



Objective Function of GAN (cont.)

▪ Objective function of generator

- 학습 초기에는 진짜 같은 가짜를 만들어낼 능력이 부족하기 때문에 D가 쉽게 가짜를 구별할 수 있음
  - 따라서 log(1−D(G(z))는 아주 작은 크기(≈0)를 가지게 되며, 결과적으로 gradient가 아주 작아지므로 학습이 거의 되지 않음

$$\min_G E_{z\sim p_z(z)}[\log(1 - D(G(z))]$$

- log(1−D(G(z))를 최소화하는 대신 log(D(G(z))를 최대화

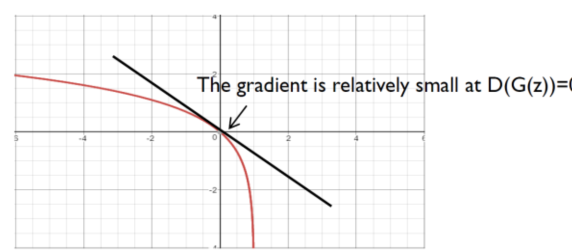
$$\cancel{\min_G E_{z\sim p_z(z)}[\log(1 - D(G(z))]}$$

Modification (heuristically motivated)

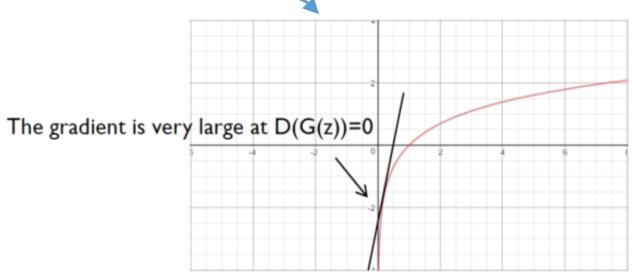
$$\max_G E_{z\sim p_z(z)}[\log D(G(z))]$$

Practical Usage

$$\min_G E_{z\sim p_z(z)}[-\log D\big(G(z)\big)]$$



$y = \log(1 - x)$



$y = \log(x)$

Training of GAN

D 학습

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

end for

G 학습

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Why does GAN work?

Because it actually minimizes the distance between the real data distribution and the model distribution.

$$\min_G \max_D V(D, G) \xrightarrow{\text{same}} \min_{G,D} JSD(p_{data} || p_g)$$

Objective function of GANs

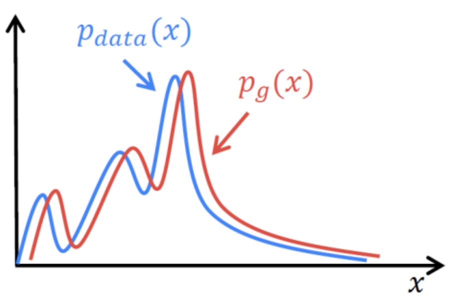
$$E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

Jenson-Shannon divergence

$$JSD(P||Q) = \frac{1}{2} KL(P||M) + \frac{1}{2} KL(Q||M)$$

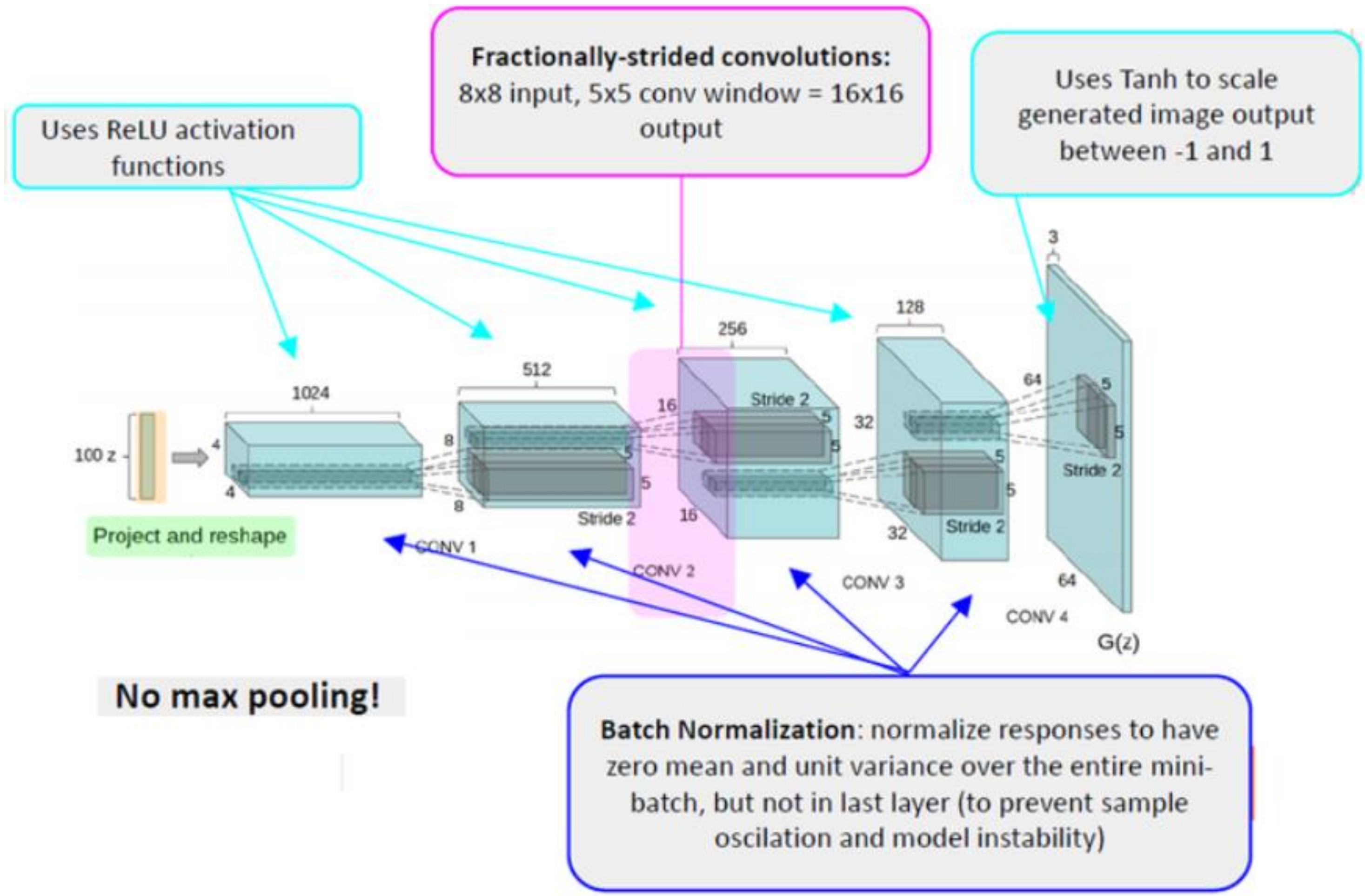
$$where\ M = \frac{1}{2}(P + Q)$$

KL Divergence



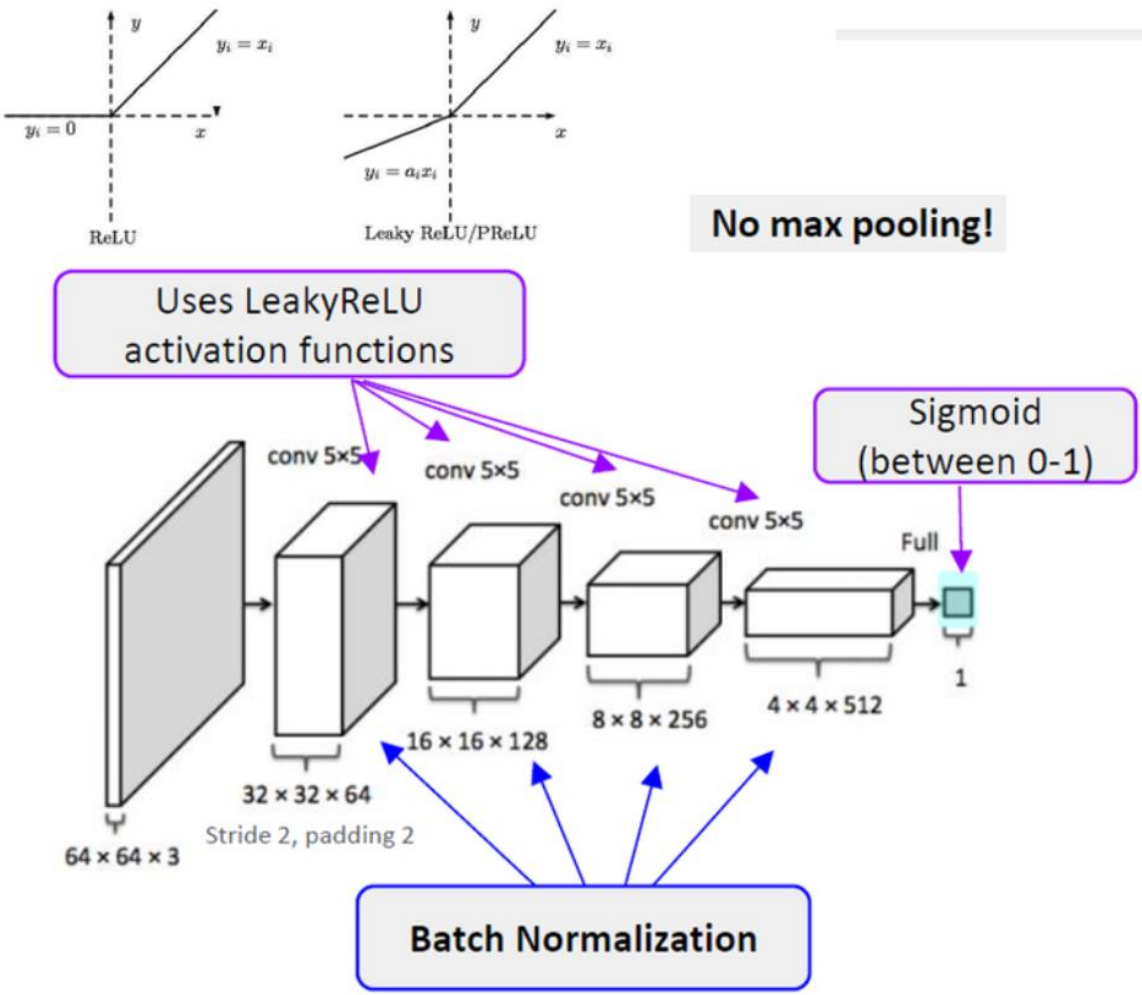
DCGAN

- Generator



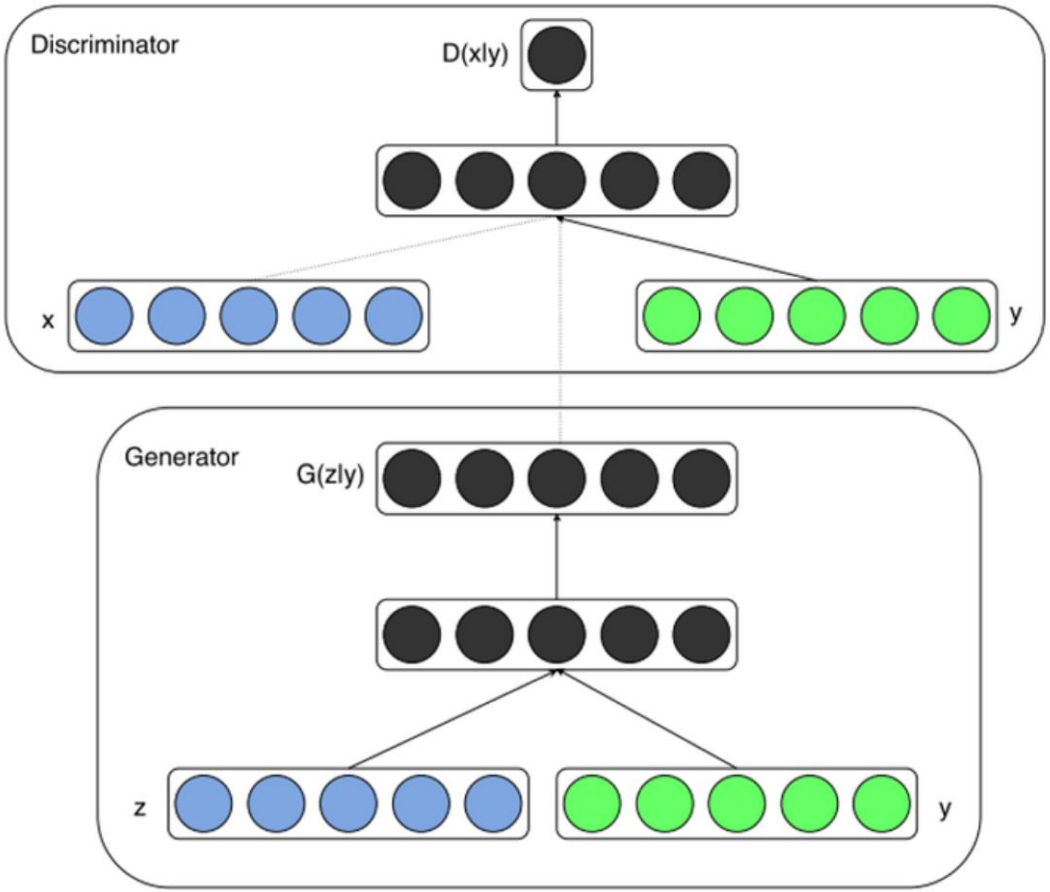
DCGAN (cont.)

- Discriminator



cGAN (Conditional GAN)

■ Architecture



y: condition
다양한 형태를 가질 수 있음 (one-hot encoding, image, ...)

KYONGGI UNIVERSITY

cGAN (Conditional GAN) (cont.)

- Loss function of original GAN

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

Expectation

x is sampled from real data

Probability of D(real)

z is sampled from N(0, 1)

Probability of D(fake)

fake

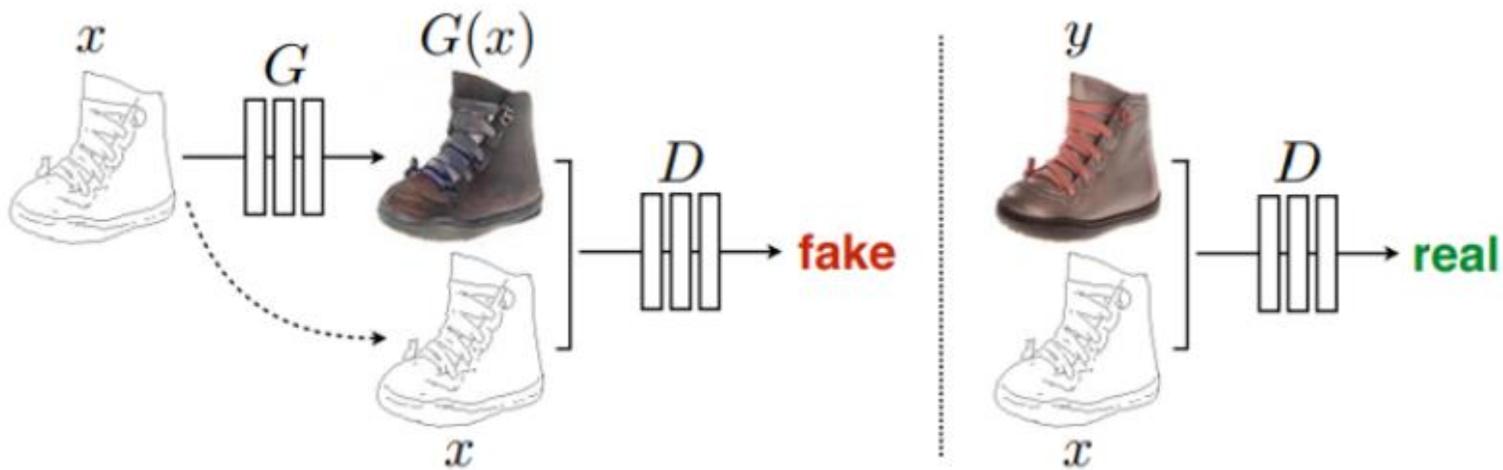
- Loss function of cGAN

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

Pix2Pix

■ Basic Idea

- Based on cGAN
- 학습 시 원영상 x와 변환하고자 하는 목표 영상 y의 쌍 (x,y)가 필요



Pix2Pix (cont.)

■ Loss function

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))]$$

- Original cGAN

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

- Reconstruction loss

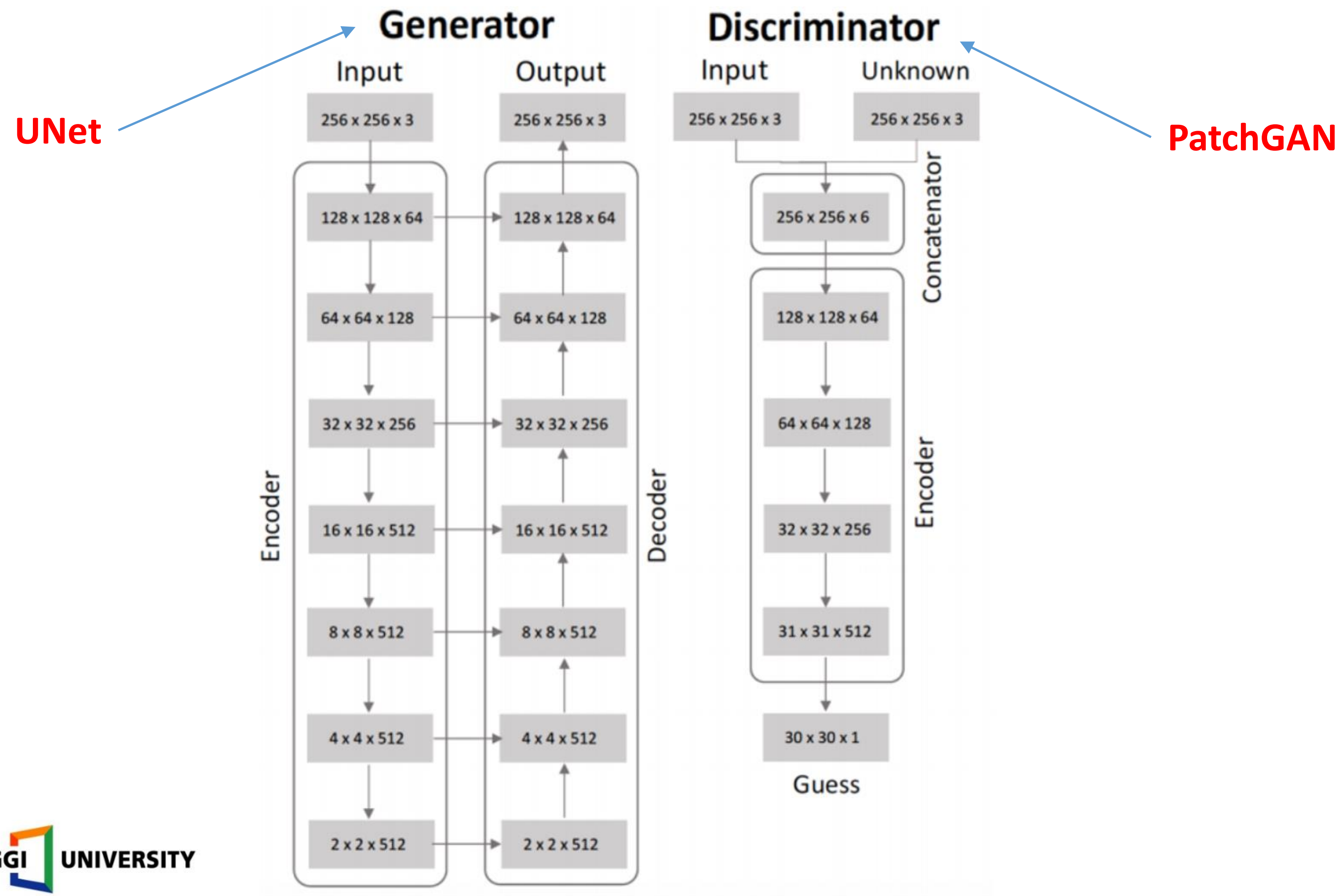
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1].$$

- Final loss function

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

Pix2Pix (cont.)

■ Network Architecture



Pix2Pix (cont.)

■ PatchGAN

Generator의 최적화 목표 함수에는 L1 loss, 즉 reconstruction loss가 포함이 되어 있는데 reconstruction loss는 원영상과 생성영상 사이의 유클리드 거리를 최소화 하는 방향이기 때문에 통상적으로 영상의 평균 성분, 즉 저주파에 집중하는 경향이 있다.

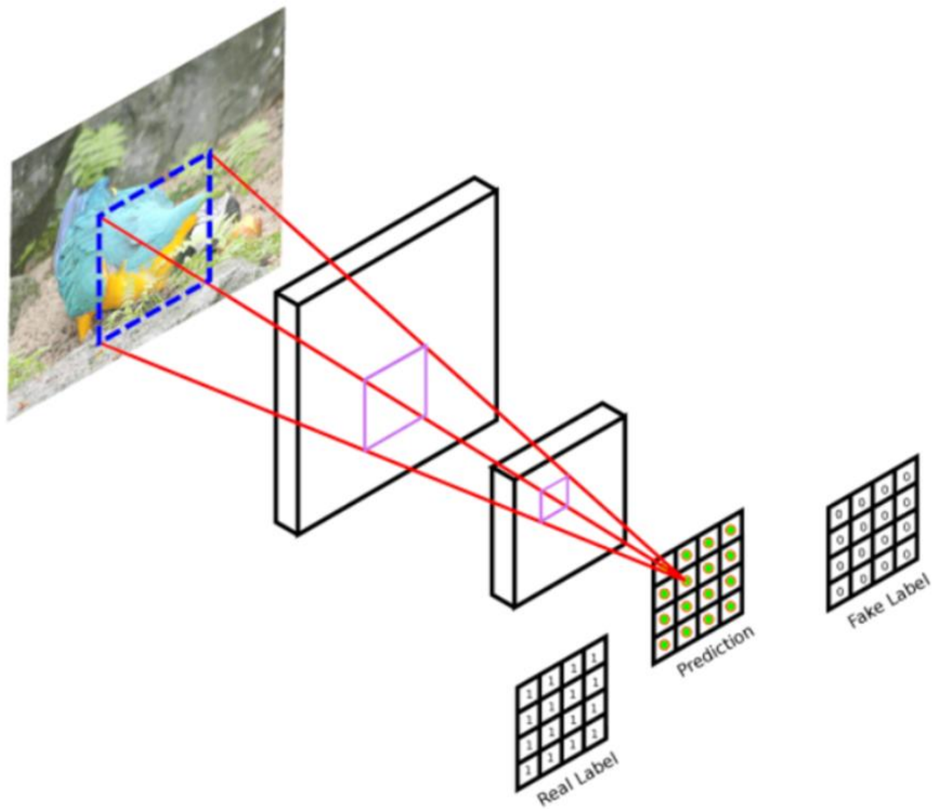
L1 loss가 저주파 영역을 감당하기 때문에 Discriminator에서는 고주파 성분(detail)에 집중하여, 진짜(real)/가짜(fake) 여부를 판단하면 된다. 통상적인 GAN 구조에서는 영상 전체에 대하여 score를 구하는 방식이었다. 원 논문에서는 이것을 ImageGAN이라고 불렀다.

↓

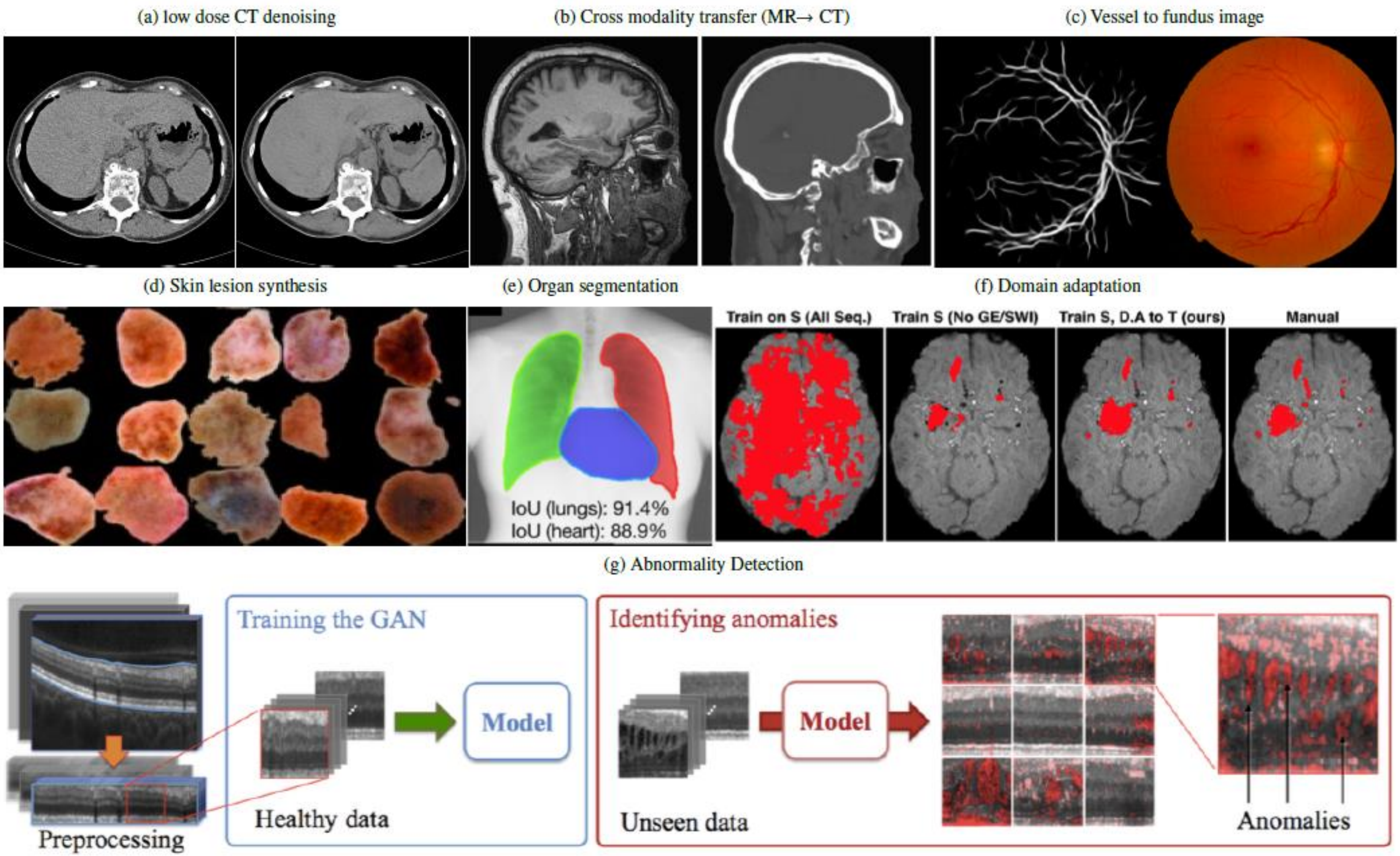
전체 영역이 아니라, 특정 크기의 patch 단위로 진짜/가짜를 판별하고, 그 결과에 평균을 취하는 방식이 PatchGAN이다. 픽셀들 간의 연관성(correlation)은 거리에 비례하여 작아지는 경향이 있으며, 일정한 거리를 넘어서게 되면 상호 간에 별 의미가 없다. 그러므로 특정 크기의 patch에 대하여 진짜 같은 이미지를 생성할 수 있다면, 그리고 그런 patch의 수가 많아지는 방향으로 학습을 하게 된다면, generator의 성능은 더 올라갈 수 있을 것이라는 사실을 추정할 수 있다.

Pix2Pix (cont.)

- PatchGAN (cont.)



GANs in Medical Domain



References

- Generative Adversarial Networks, https://arxiv.org/abs/1406.2661
- Conditional Generative Adversarial Nets, https://arxiv.org/abs/1411.1784
- Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, https://arxiv.org/abs/1511.06434
- Image-to-Image Translation with Conditional Adversarial Networks, https://arxiv.org/abs/1611.07004
- Generative Adversarial Network in Medical Imaging: A Review, https://arxiv.org/abs/1809.07294
- 라온피플 머신러닝 아카데미, https://laonple.blog.me/


ANY
QUESTIONS?