Generative Adversarial Network
: Image-to-Image Translation Pix2Pix

Seokkyeong Boo

Image-to-Image Translation : Pix2Pix

➢ 대표적인 Image-to-Image Translation 모델.

- ➢ Pix2Pix는 학습 과정에서 이미지 자체를 조건(Condition)으로 입력 받는 cGAN의 한 유형이다.
- ➢ Pix2Pix는 픽셀(Pixel)들을 입력으로 받아 픽셀(Pixel)들을 예측한다는 의미의 이름이다.

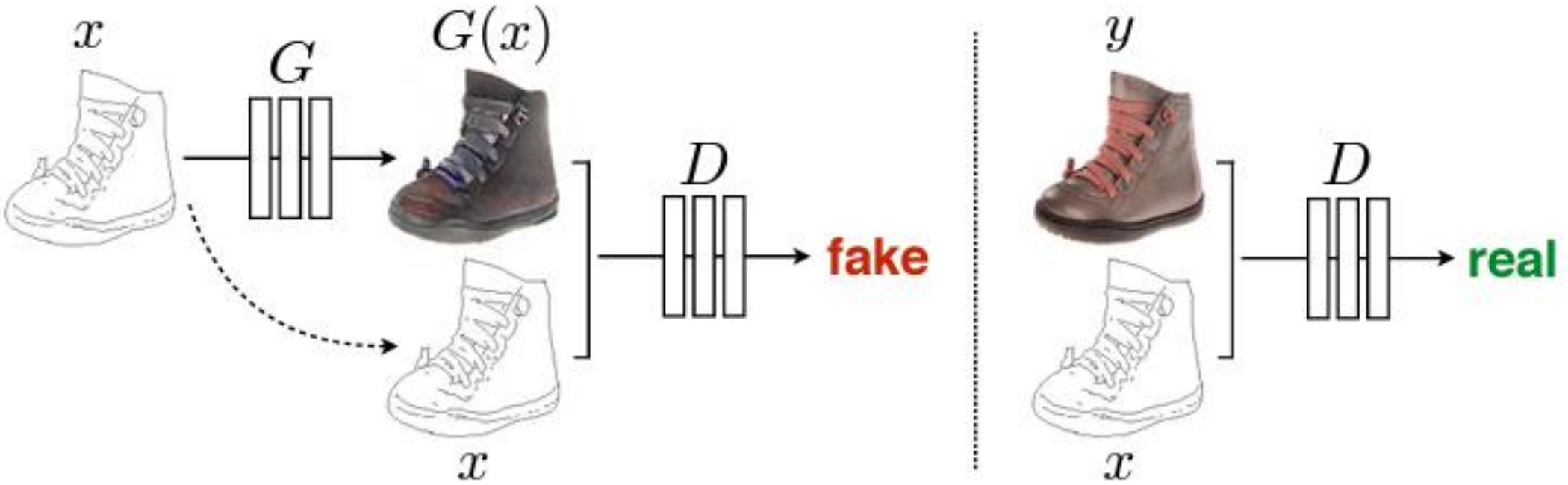


Image-to-Image Translation : Pix2Pix

- 이미지를 조건(Condition)으로 입력 받고, 이미지를 출력으로 내보낸다.
- 이를 효과적으로 처리할 수 있는 U-Net 기반의 네트워크 구조를 사용한다.
  - 입력과 출력의 차원을 같게 하는 구조는 다양하지만, 본 논문에서는 U-Net구조를 사용했을 때 비교적 좋은 성능을 나타냄.

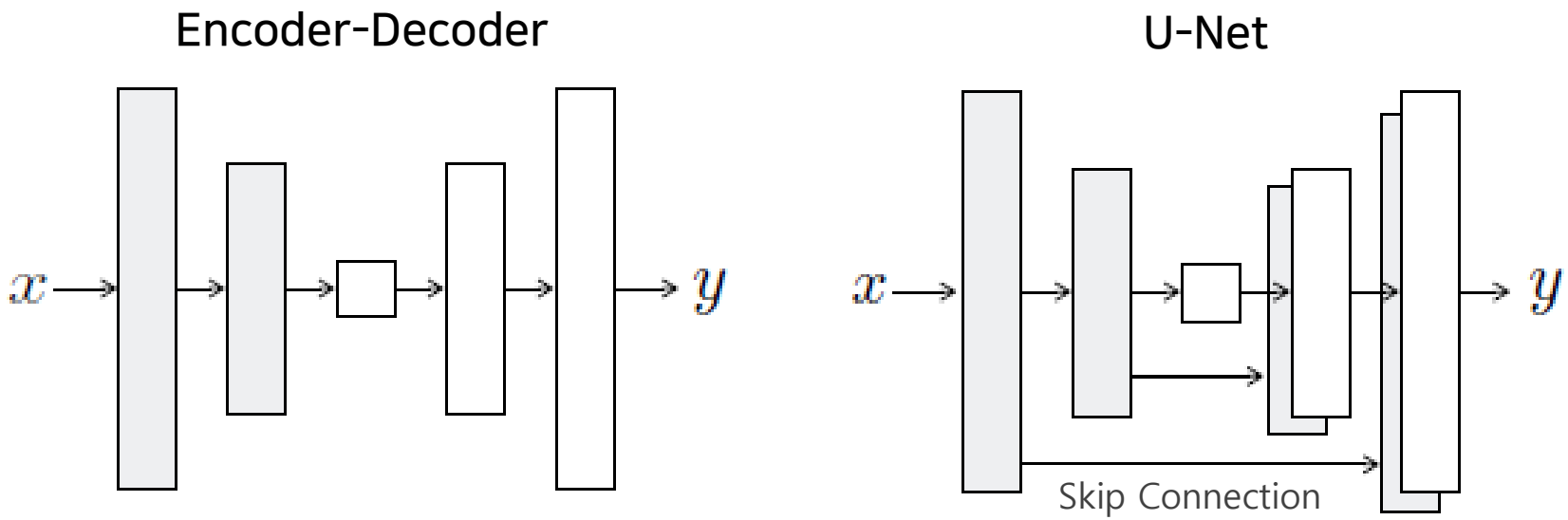


Image-to-Image Translation : Pix2Pix

- Encoder-Decoder
  - 간단하게 Encoder-Decoder 구조를 이해하기 위해서 Pictionary 게임을 볼 수 있다.
  - 플레이어1은 목록에서 무작위 단어를 선택하고 단어에 맞는 스케치를 한다.
  - 플레이어2는 그림을 분석하고 설명하는 단어를 식별하게 된다.

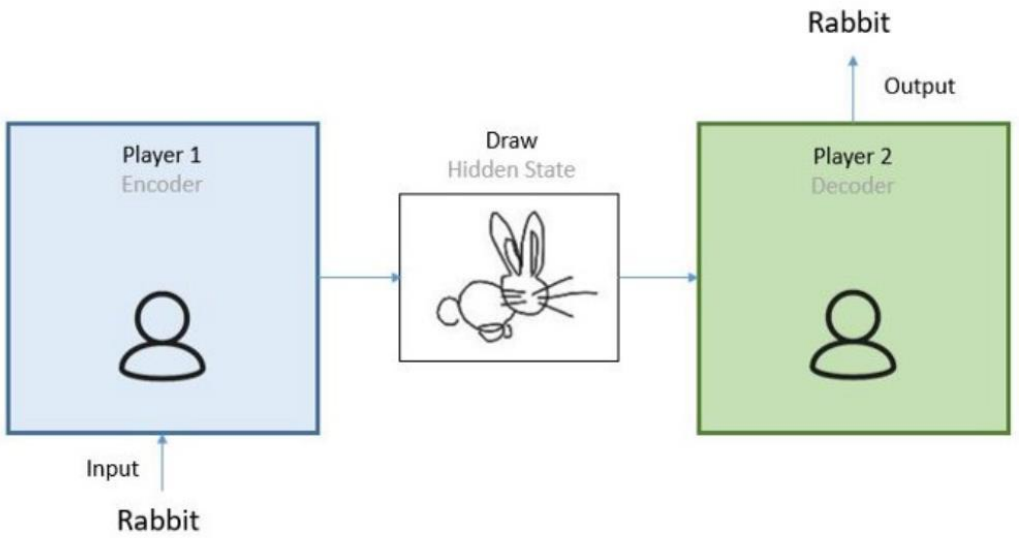


Image-to-Image Translation : Pix2Pix

- 이미지를 조건(Condition)으로 입력받고, 이미지를 출력으로 내보낸다.
- 이를 효과적으로 처리할 수 있는 U-Net 기반의 네트워크 구조를 사용한다.
  - U-Net의 Skip connection은 Encoder에서 사용된 출력정보들을 Decoder에서 사용함.
  - 입력과 출력이 동일한 구조를 가지고 이미지를 만들고 공유되는 정보가 많아 U-Net이 좋은 성능을 나타내는 것이라고 판단.

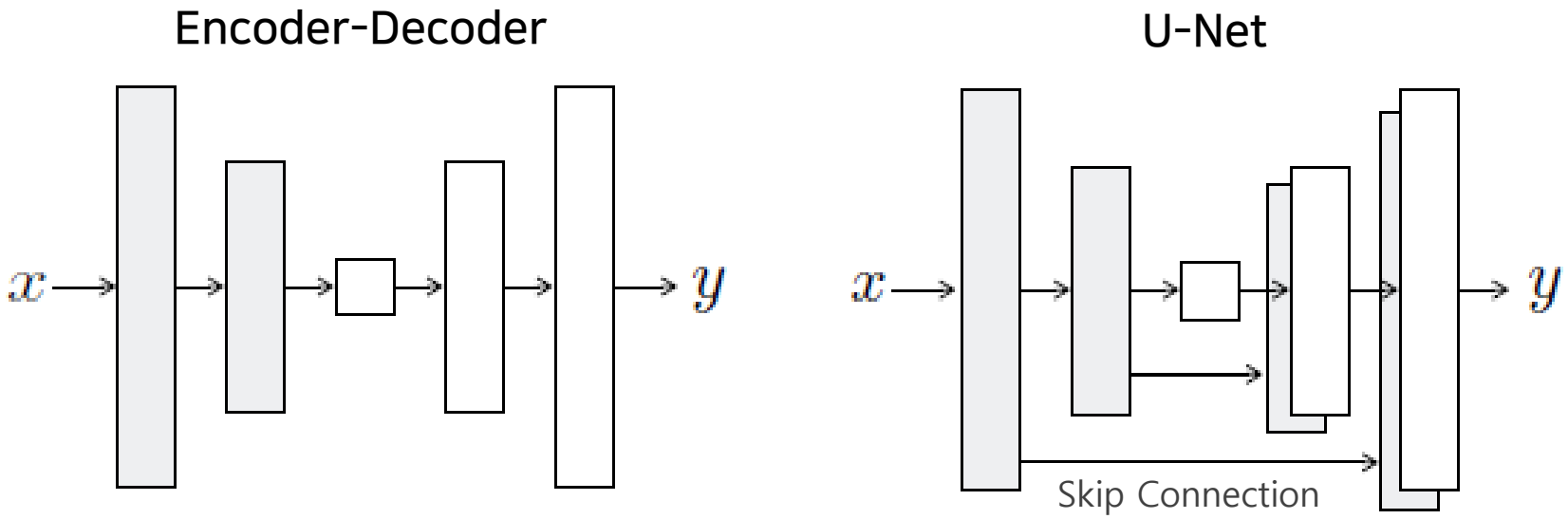


Image-to-Image Translation : Pix2Pix

- GAN은 기본적으로 다른 생성 모델에 비해 흐릿한(blurry) 결과가 나오는 문제가 적은 편이다.
- GAN의 성능을 더 향상시키기 위해 L1 Loss 함수를 함께 사용한다. (ground-truth와 유사한 결과)
  - 유클리드 거리(Euclidean distance)는 결과들의 평균값 자체가 낮아지는(minimized)방향으로 구성되어, 흐릿한 결과가 나올 수 있다.

목적 함수 : $$G^* = \arg\min_G \max_D Loss_{cGAN}(G,D) + Loss_{L1}(G)$$

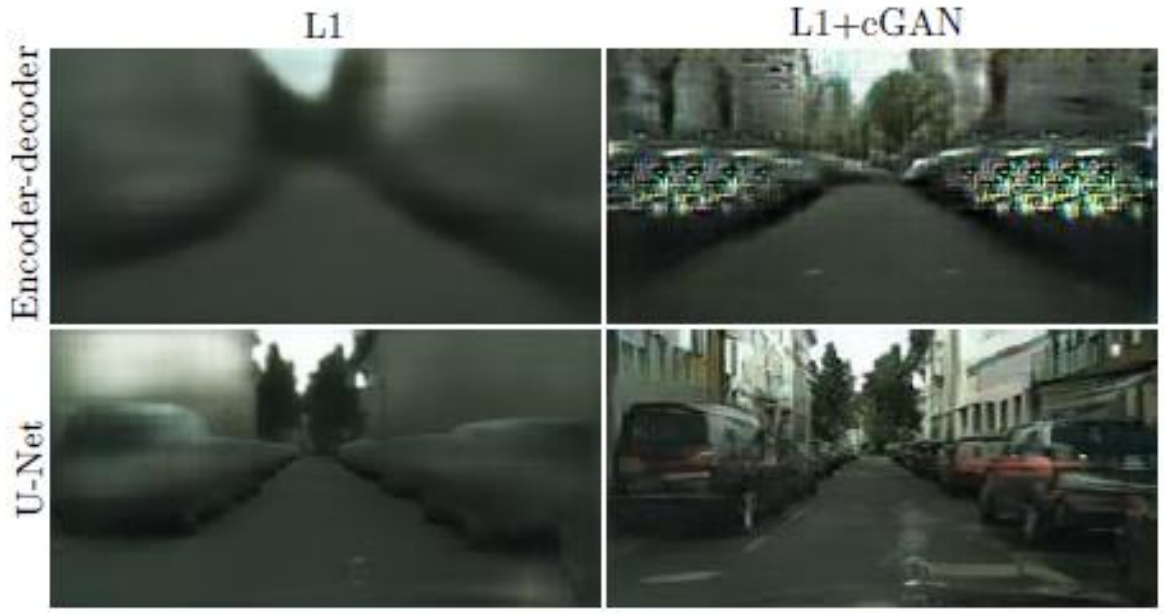
$$Loss_{cGAN}(G,D) = \mathrm{E}_{x,y}\left[\log D(x,y)\right] + \mathrm{E}_{x,z}\left[\log\left(1 - D\left(x, G(x,z)\right)\right)\right]$$

$$Loss_{L1}(G) = \mathrm{E}_{x,y,z}\left[\left\| y - G(x,z) \right\|_1\right]$$

Image-to-Image Translation : Pix2Pix

- Pix2Pix 식별자(Discriminator)는 Convolution PatchGAN 분류 모델을 사용한다.
  - 이미지 전체에 대하여 판별하지 않고, 이미지 내 패치 단위로 판별한다.

- 성능 평가



Adding skip connections to an encoder-decoder to create a "U-Net" results in much higher quality results.

Image-to-Image Translation : Pix2Pix

Figure 4: Different losses induce different quality of results. Each column shows results trained under a different loss. Please see https://phillipi.github.io/pix2pix/ for additional examples.

Image-to-Image Translation : Pix2Pix

➢ Generator architectures

- ➢ Ck : Convolution – BatchNorm – ReLU층에 k개의 필터가 있다고 정의.
- ➢ CDk : Convolution – BatchNorm – Dropout - ReLU 층에 k개의 필터가 있다고 정의. (dropout rate는 50%)
- ➢ 모든 Convolution layer는 4 x 4 filter 에 stride 2.

- ➢ 마지막 Decoder에서 output channels을 3개의 channels로 설정, Tanh function.
- ➢ BatchNorm은 Encoder의 첫번째 C64에서는 미적용.
- ➢ Encoder의 ReLU 는 slope가 0.2인 leaky ReLU, Decoder는 slope가 0인 ReLU적용

- ➢ Encoder
  - ✓ C64 – C128 – C256 – C512 – C512 – C512 – C512 – C512
- ➢ Decoder
  - ✓ CD512 – CD512 – CD512 – C512 – C256 – C128 – C64
- ➢ U-Net Decoder
  - ✓ CD512 – CD1024 – CD1024 – C1024 – C512 – C256 – C128

Image-to-Image Translation : Pix2Pix

- Generator Network

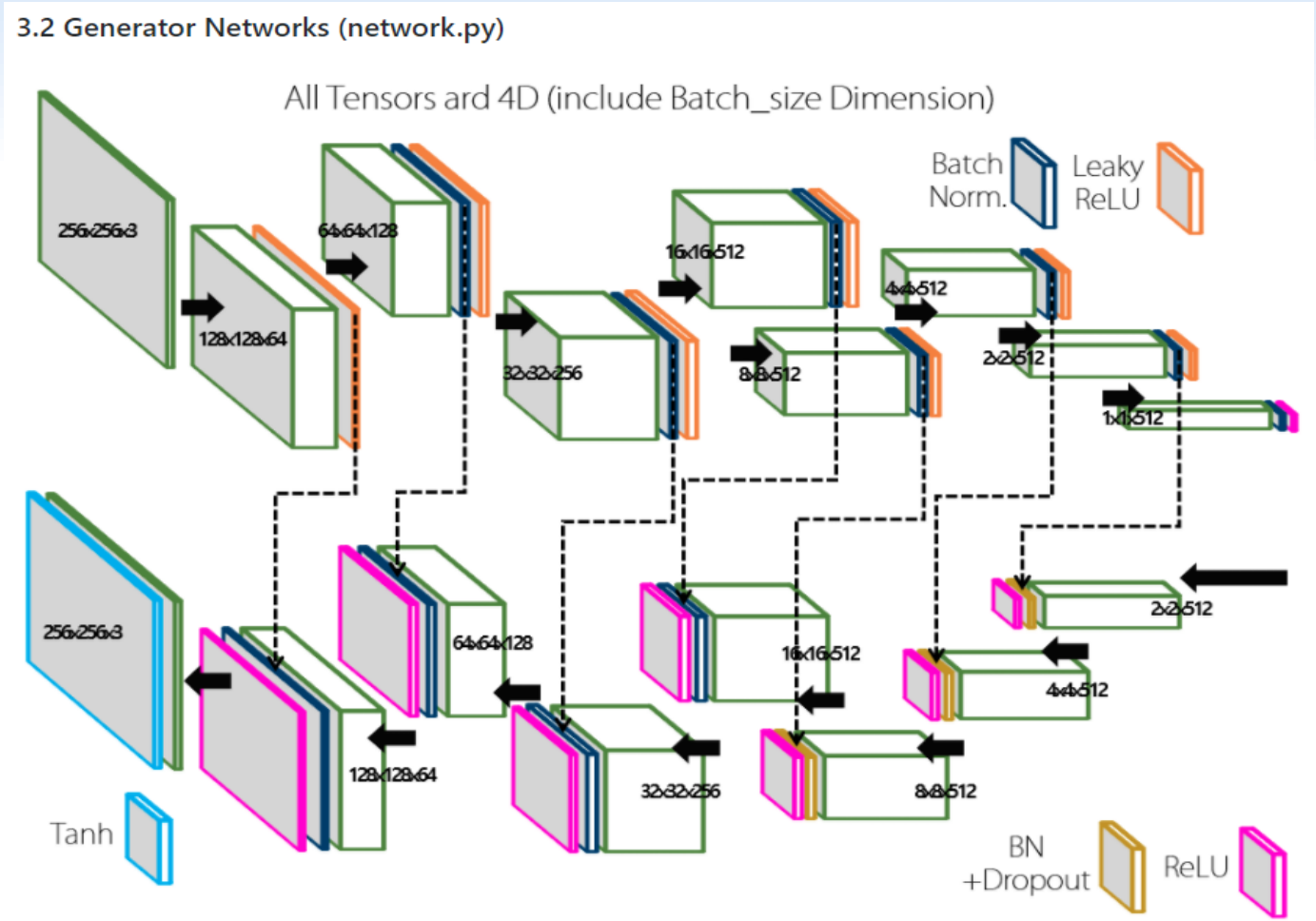


Image-to-Image Translation : Pix2Pix

- Discriminator architectures
  - C64 – C128 – C256 – C512

  - 이후에 1차원의 output으로 설정.
  - Sigmoid function.
  - 첫번째 C64 층에 BatchNorm 적용하지 않음.
  - 모든 ReLU는 slope가 0.2인 leaky ReLU 적용.
  - Patch Gan 을 이용함.
  - 입력으로 두개의 이미지를 받고 결합하여 분류

- Training details
  - Jittering
    - ✓ 입력 이미지(256 x 256)를 resizing(286 x 286)한 후 다시 랜덤하게 Cropping(256 x 256).

Image-to-Image Translation : Pix2Pix

➢ Discriminator Network

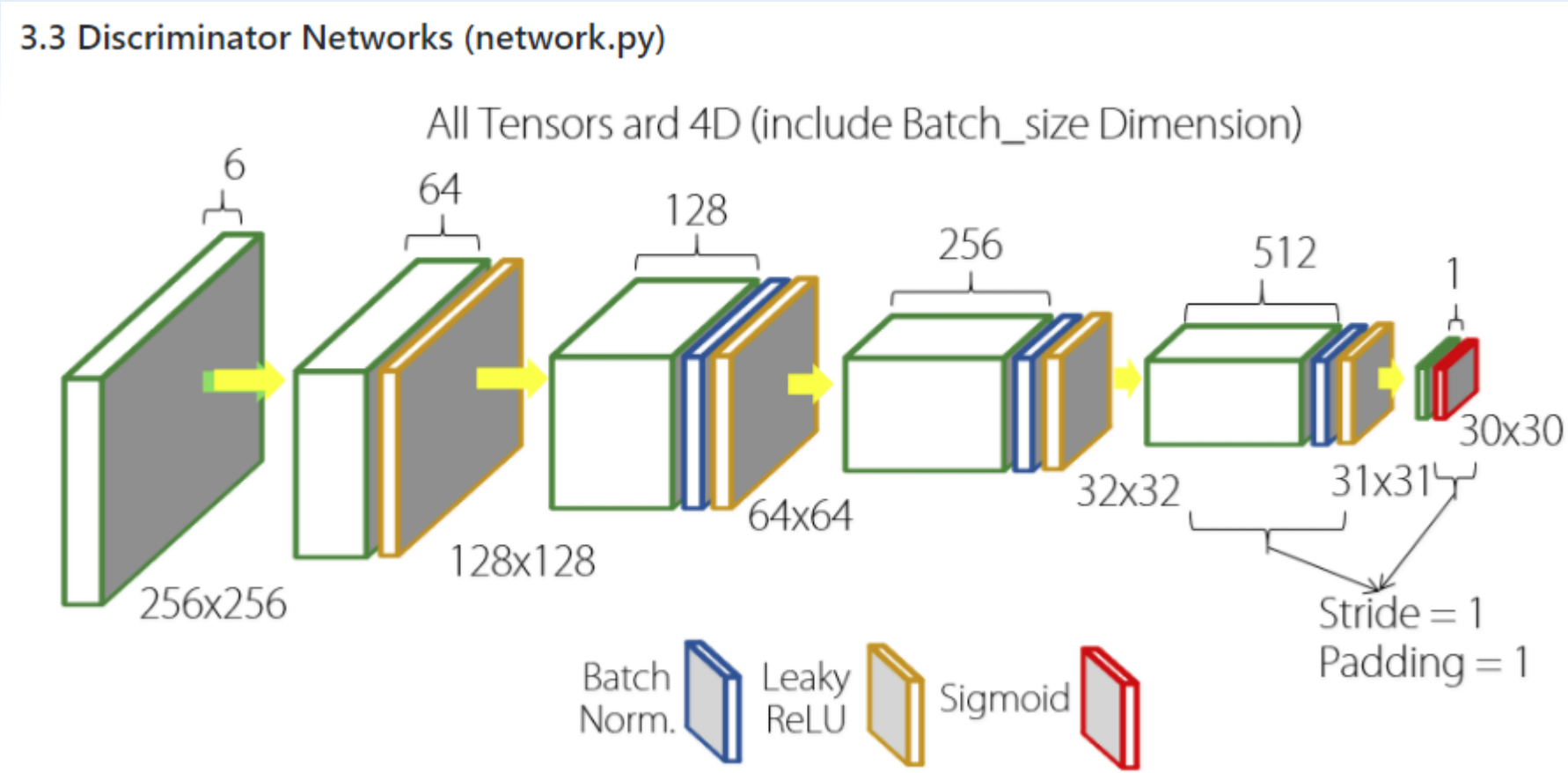


Image-to-Image Translation : Pix2Pix

- PatchGAN
  - 이미지에서 특정 크기의 patch 를 가지고 이미지의 참과 거짓을 판단한다.
  - 전체 이미지 크기에서 픽셀 간의 관계를 파악하여 적절한 크기의 patch 사이즈를 정한다.
  - Pix2Pix 모델에서 patch 사이즈는 70 * 70이다.
  - 좌측의 특징 맵의 모든 값을 평균을 출력한다.

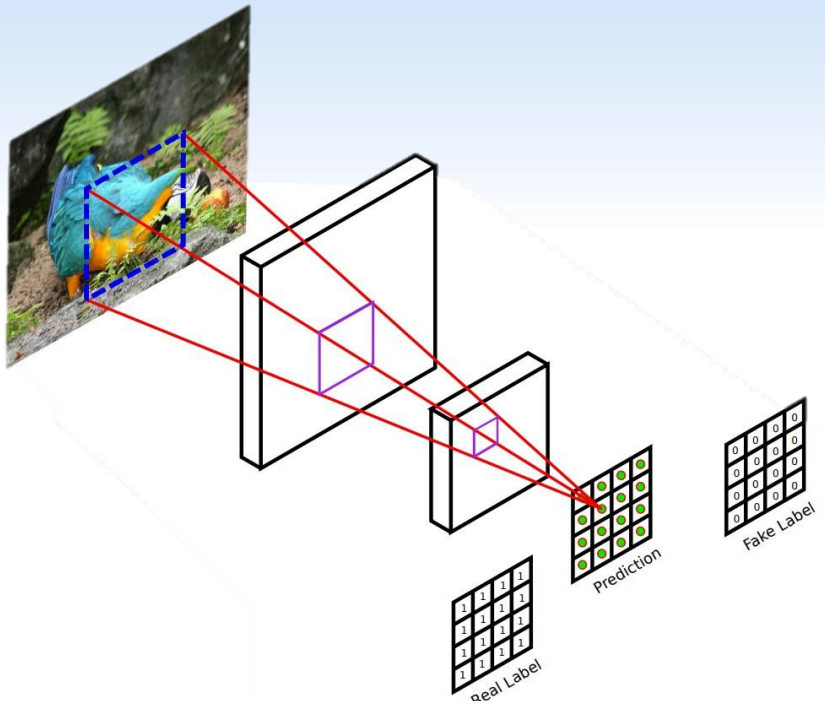
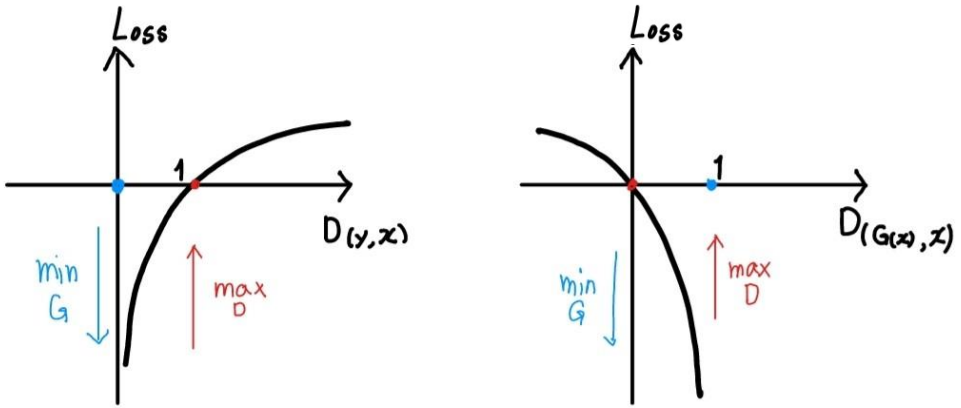


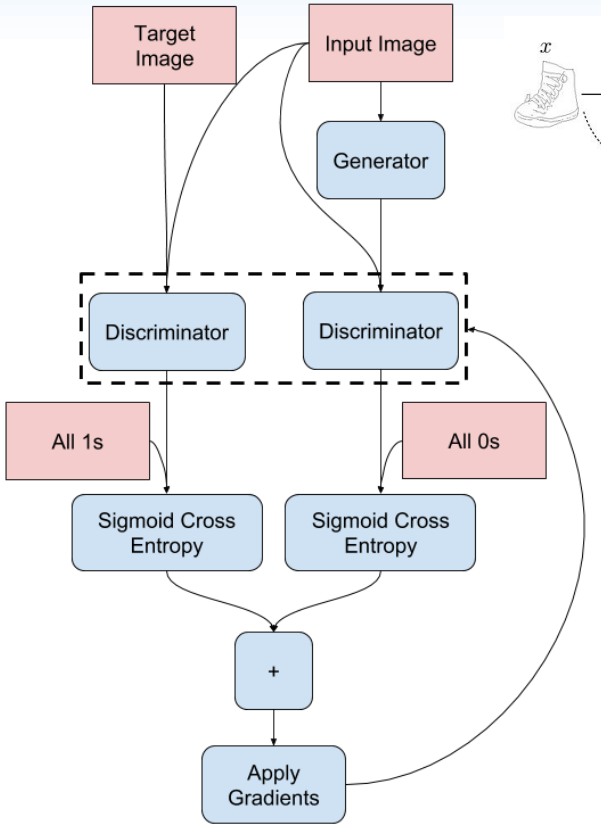
Image-to-Image Translation : Pix2Pix

➢ 손실함수

➢ 판별자의 손실함수 : 실제 이미지로부터의 Loss값과 생성된 이미지로부터의 Loss값의 합으로 나타낸다.

$$Loss_D = \mathrm{E}_{x,y}\left[\log D(y,x)\right] + \mathrm{E}_{x,z}\left[\log\left(1 - D\left(G(x),x\right)\right)\right]$$





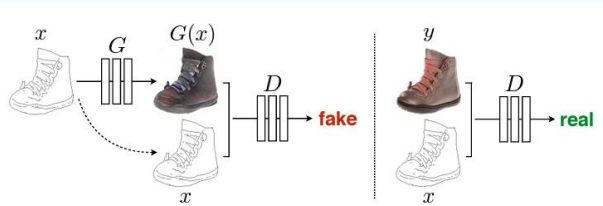


Image-to-Image Translation : Pix2Pix

➢ 손실함수

- ➢ 생성자의 손실함수 : 생성된 이미지로부터의 Loss값과 생성된 이미지와 타겟 이미지 사이에 MSE Loss값으로 이루어짐.
- ➢ 여기서 L1 Loss의 가중치를 100으로 설정함.

$$Loss_G = \mathrm{E}_{x,z}\left[\log\left(D\left(G\left(x\right),x\right)\right)\right] + \lambda E_{x,y,z}\left[\left\|y - G\left(x\right)\right\|_1\right]$$

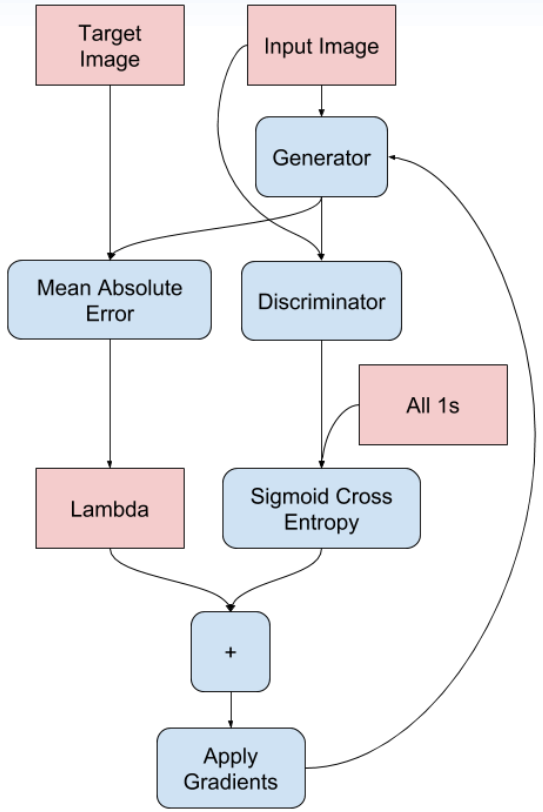


Image-to-Image Translation : Pix2Pix

- 최적화 기법
  - Adam
    - ✓ 학습률 : 0.0002
    - ✓ α_1 : 0.5, α_2 : 0.999

알고리즘 5–5 Adam

입력: 훈련집합 $\mathbb{X}$, $\mathbb{Y}$, 학습률 ρ, 모멘텀 계수 α_1, 가중 이동 평균 계수 α_2
출력: 최적의 매개변수 $\hat{\boldsymbol{\Theta}}$

1 난수를 생성하여 초기해 $\boldsymbol{\Theta}$를 설정한다.
2 $\mathbf{v} = \mathbf{0}, \mathbf{r} = \mathbf{0}$
3 t=1
4 repeat
5 그레이디언트 $\mathbf{g} = \frac{\partial J}{\partial \boldsymbol{\Theta}}\Big|_{\boldsymbol{\Theta}}$를 구한다.
6 $\mathbf{v} = \alpha_1 \mathbf{v} - (1-\alpha_1)\mathbf{g}$ // 속도 벡터
7 $\mathbf{v} = \frac{1}{1-(\alpha_1)^t}\mathbf{v}$
8 $\mathbf{r} = \alpha_2 \mathbf{r} + (1-\alpha_2)\mathbf{g} \odot \mathbf{g}$ // 그레이디언트 누적 벡터
9 $\mathbf{r} = \frac{1}{1-(\alpha_2)^t}\mathbf{r}$
10 $\Delta\boldsymbol{\Theta} = -\frac{\rho}{\epsilon+\sqrt{\mathbf{r}}}\mathbf{v}$
11 $\boldsymbol{\Theta} = \boldsymbol{\Theta} + \Delta\boldsymbol{\Theta}$
12 t++
13 until (멈춤 조건)
14 $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}$

Image-to-Image Translation : Pix2Pix

- 데이터 세트(Facades)
  - 다음과 같은 쌍으로 되어있는 데이터 세트이다.
  - 400개의 훈련 집합.
  - 100개의 validation 집합.
  - 100개의 테스트 집합.

  - validation / test
    - ✓ validation 집합은 학습이 완료된 모델을 검증하기 위한 집합이다.
    - ✓ test 집합은 학습과 검증이 완료된 모델을 평가하기 위한 집합이다.

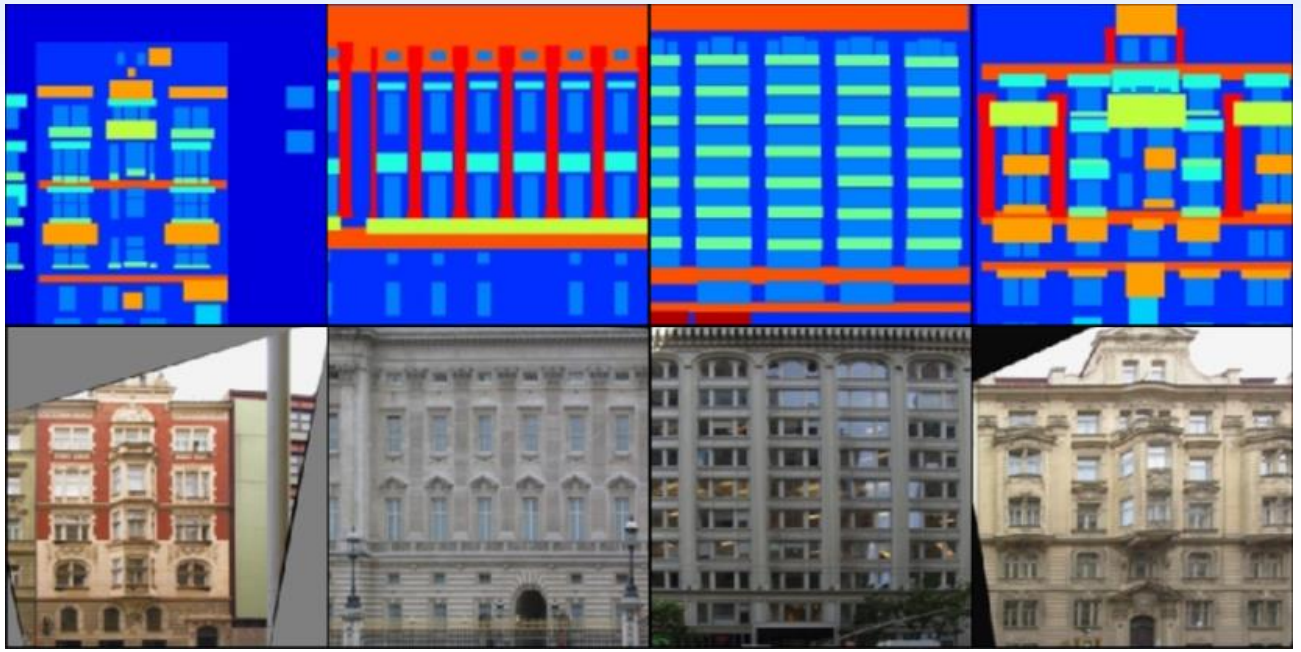


Image-to-Image Translation : Pix2Pix

- **코드 결과**
  - 학습률 : 0.0002
  - 배치 사이즈 : 4
  - 이미지 사이즈 : 256 x 256 x 3
  - 진행 Epoch : 300

  - 50epoch 동안 판별자를 먼저 학습 후 판별자와 생성자를 학습.

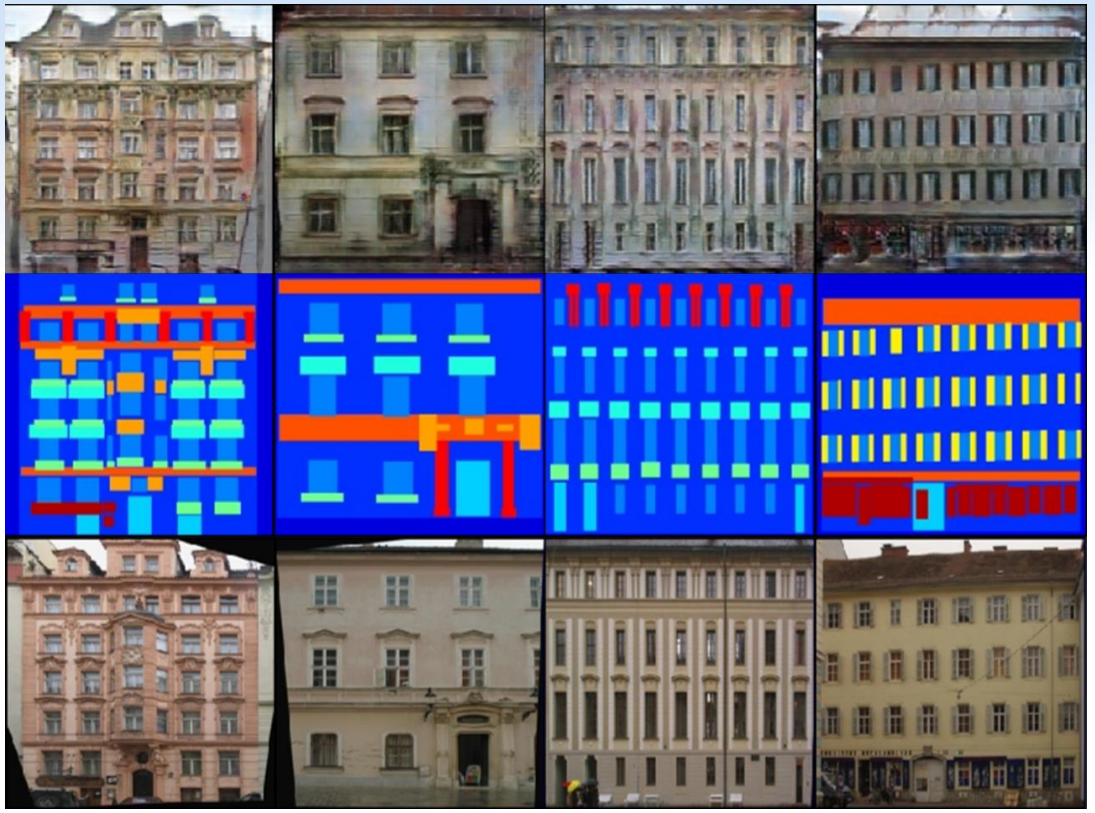
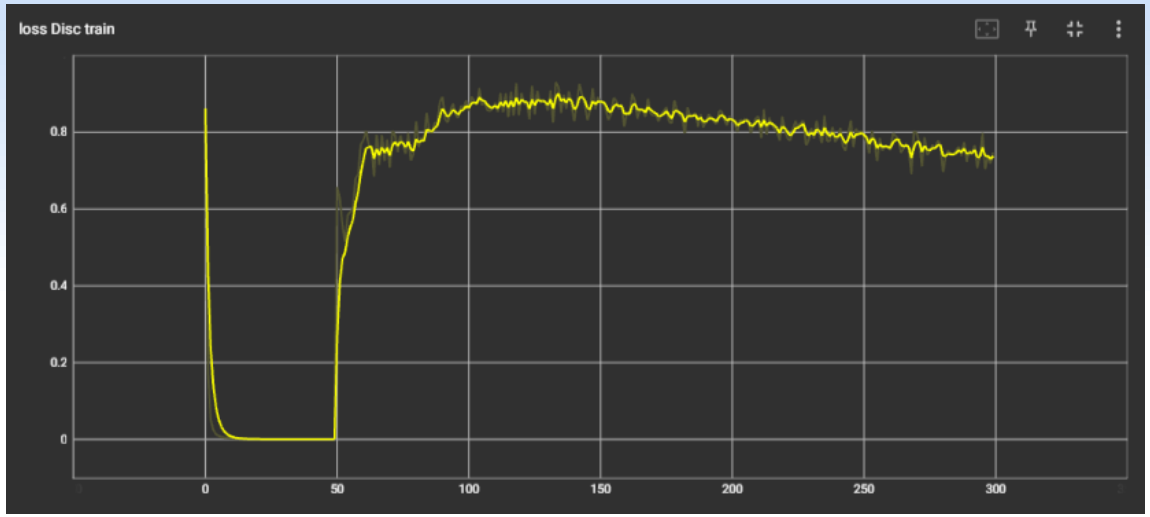


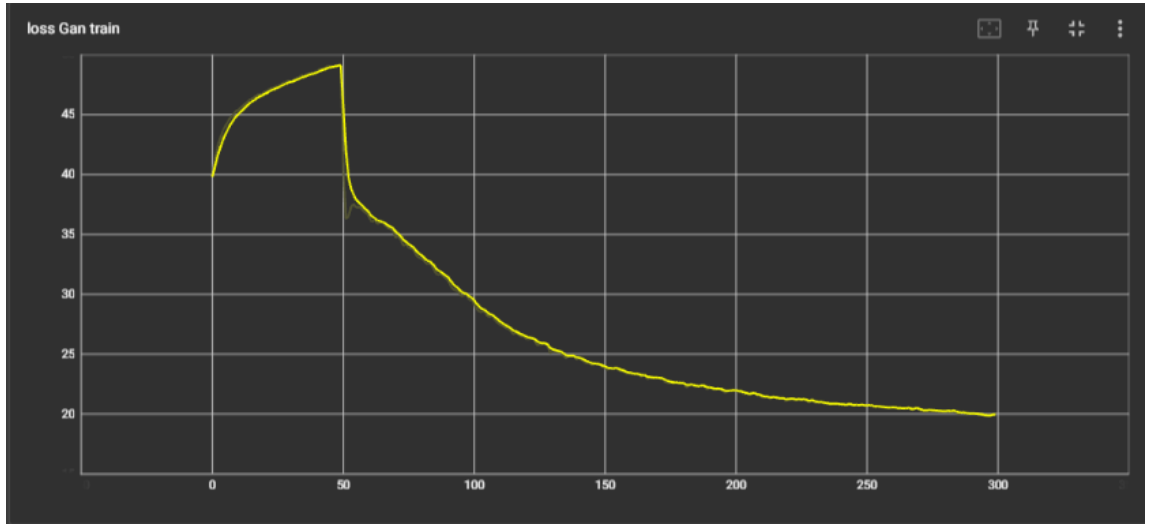
Image-to-Image Translation : Pix2Pix

➢ 코드 결과(Train Loss)

$$Loss_D = 0.5\left(\mathrm{E}_{x,y}\left[\log D(y,x)\right] + \mathrm{E}_{x,z}\left[\log\left(1 - D\left(G(x),x\right)\right)\right]\right)$$



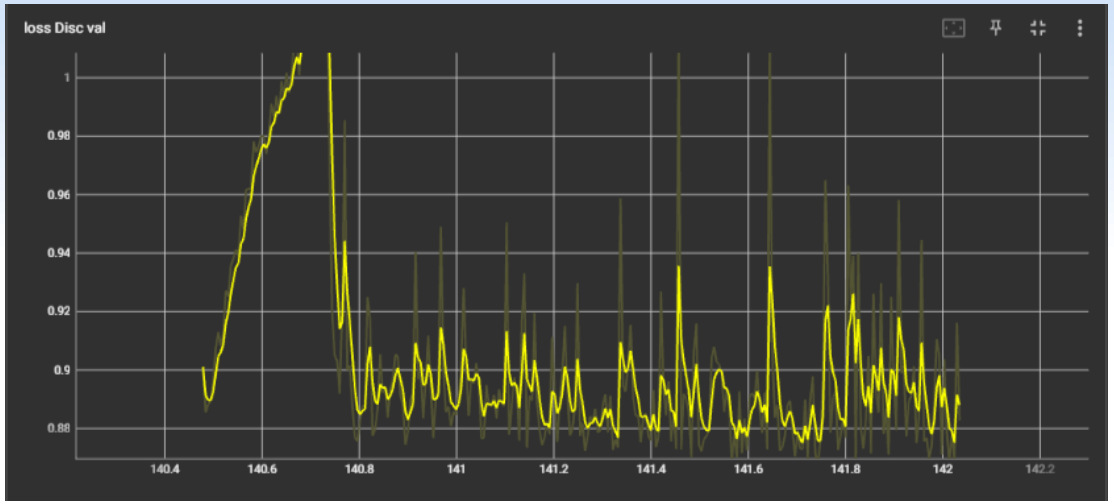
$$Loss_G = \mathrm{E}_{x,z}\left[\log\left(D\left(G(x),x\right)\right)\right] + \lambda E_{x,y,z}\left[\left\|y - G(x)\right\|_1\right]$$



where. $\lambda = 100$

Image-to-Image Translation : Pix2Pix

➢ 코드 결과(Validation Loss)



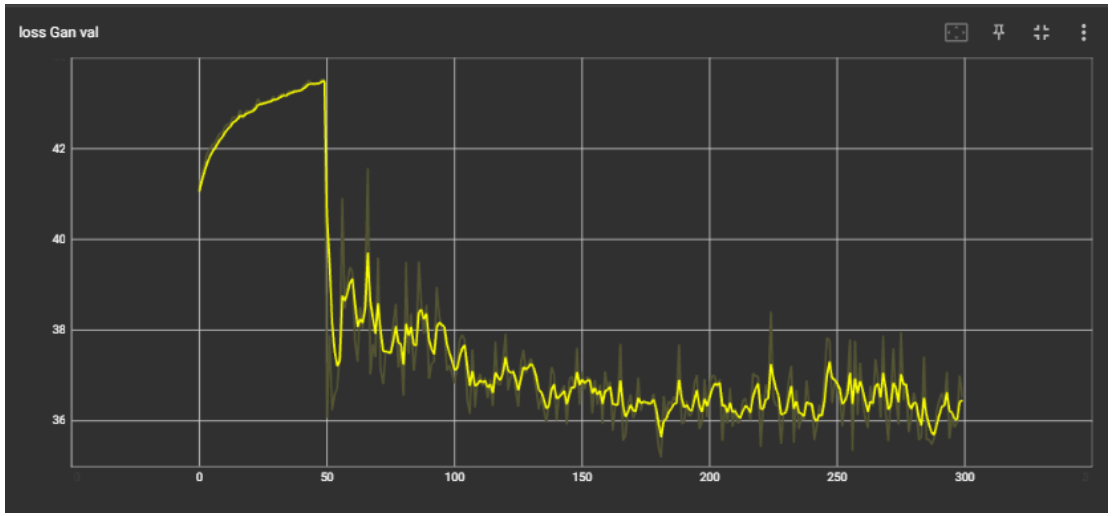


Image-to-Image Translation : Pix2Pix

- 코드 결과(Test)
  - Generator Loss : 1.3264
  - L1 loss : 0.3552
  - real image Discriminator loss : 0.9085
  - fake image Discriminator loss : 0.4148