pix2pix

YBIGTA 18기 남종현

Contents

GAN / cGAN

- GAN

Generator와 Discriminator 두 개의 네트워크를 활용한 생성 모델

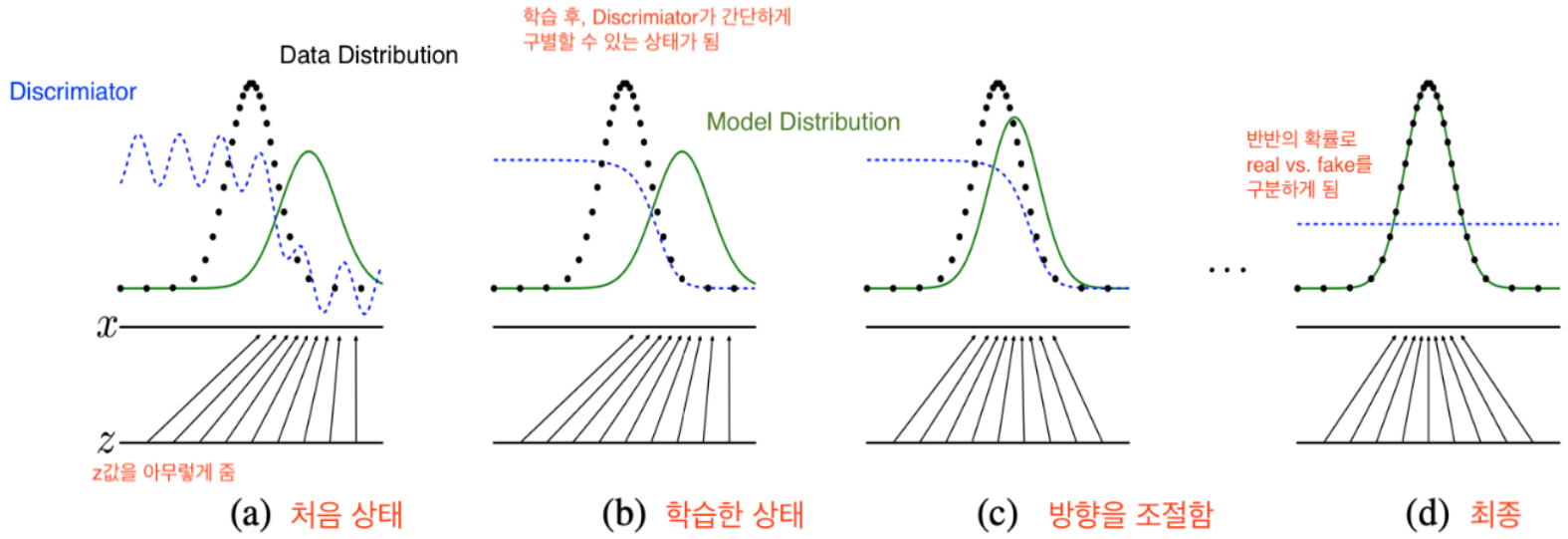
G: 실제 데이터와 유사한 데이터를 만들어 D를 속이는 방향으로 학습

D: real = 1, fake = 0으로 구분하도록 학습

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

GAN / cGAN

- GAN



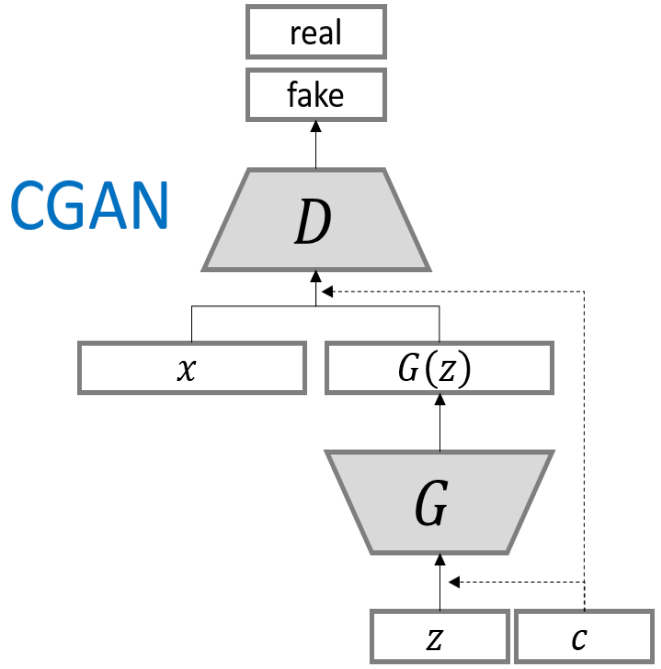
G가 D 분포를 완벽한 수준으로 복원했을 때, 구분할 확률은 1/2

GAN / cGAN

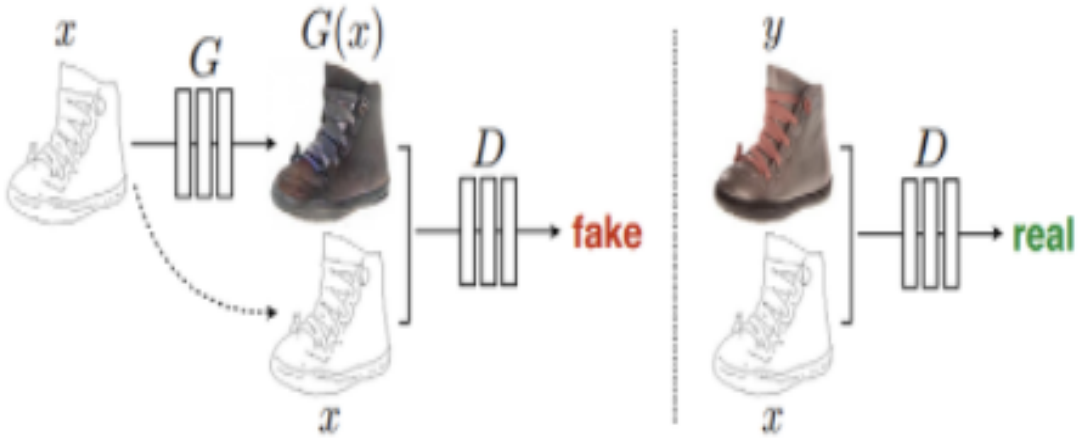
- **cGAN**

Unconditional 생성모델에서 데이터가 생성되는 종류를 제어할 방법은 없음

조건 설정(one-hot vector 등)을 통해 데이터 생성 과정을 제어할 수 있음



pix2pix Objective



Input vector x 와 noise vector z를 이용해 output vector y를 생성

Input image의 distribution 정보도 real/fake를 구별하는 데 사용

pix2pix Objective

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

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

cGAN Loss(adversarial Loss) + L1 Loss(reconstruction Loss)

Our final objective is

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G). \qquad (4)$$

pix2pix architecture

-Generator

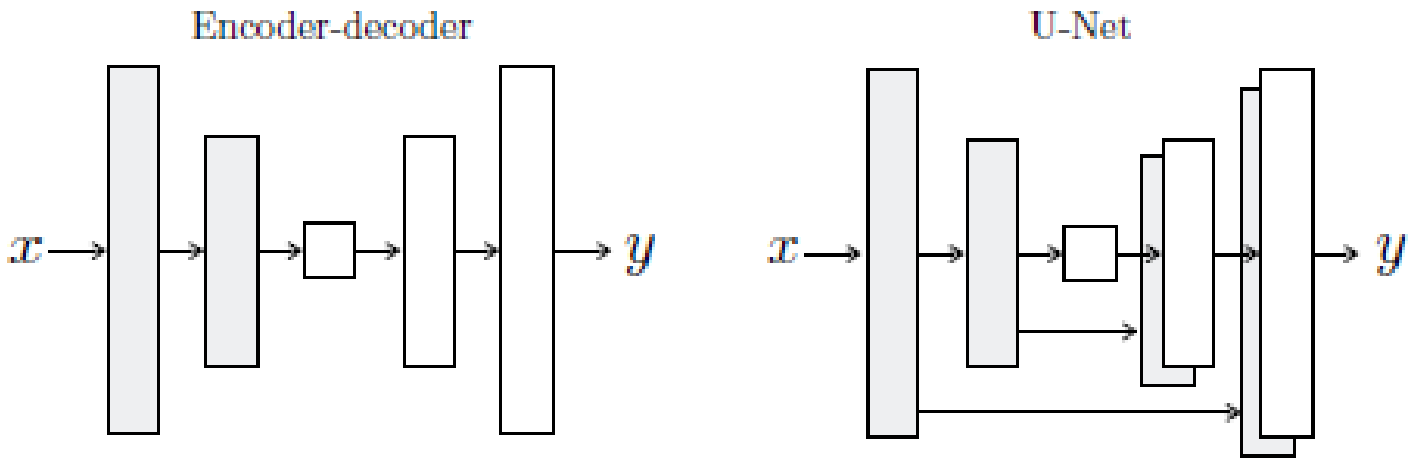


Figure 3: Two choices for the architecture of the generator. The "U-Net" [50] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

pix2pix architecture

-Generator

Encoder-decoder 구조를 사용할 때, 정보의 손실이 발생

Skip-connection을 추가한 U-Net 구조 사용

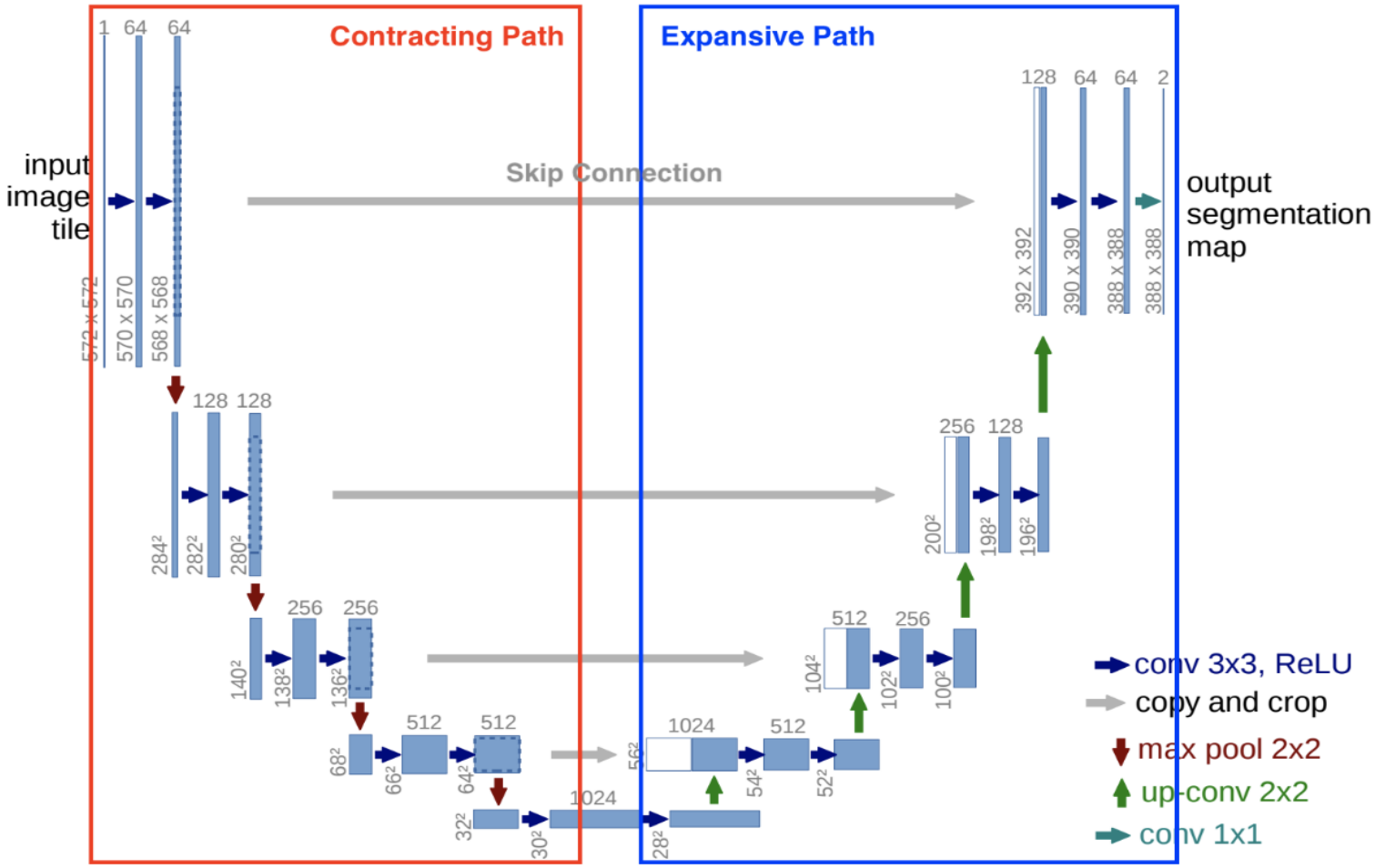
전체 Layer 수를 N이라고 할 때, i번째 layer와 N-i번째 layer를 연결

각 연결은 단순히 concatenate 한 것

처음 detail들이 마지막 layer까지 잘 전달됨

pix2pix architecture

-Generator



pix2pix architecture

-Generator layers

Encoder: C64 - C128 - C256 - C512 – C512 – C512 – C512

U-Net decoder: CD512 – CD1024 – CD1024 – C1024 – C1024 – C512 - C256 – C128

- Decoder의 마지막 layer에서 output channel(3)으로 mapping, 이후 tanh 사용
- Encoder의 첫번째 layer에서 BatchNorm 적용 x
- Encoder: leaky ReLU(slope=0.2), decoder:ReLU 사용

- Ck: Conv-BatchNorm-ReLU layer
- CDk: Conv-BatchNorm-Dropout-ReLU layer with a dropout rate of 50%
- All conv는 4x4 spatial filter, stride=2
- Downsample, upsample by a factor 2

pix2pix architecture

-Discriminator

PatchGAN 구조 사용

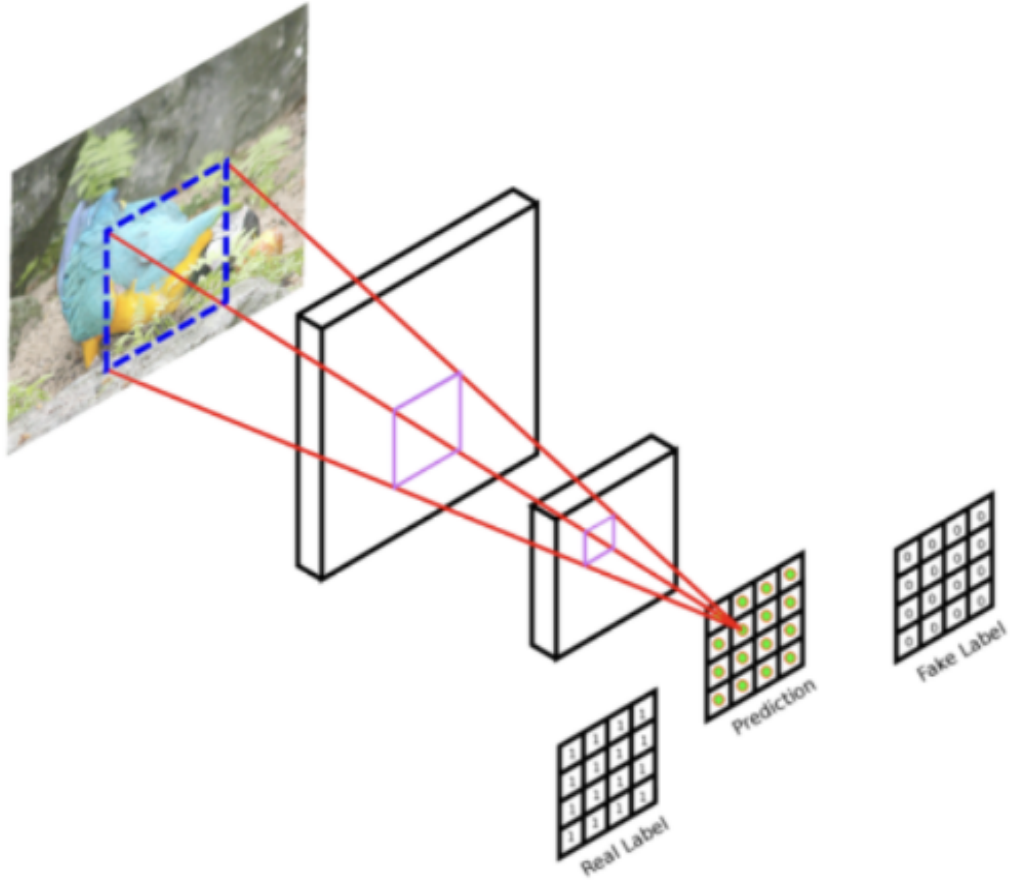
N x N 의 각 patch 별로 real/fake 판별

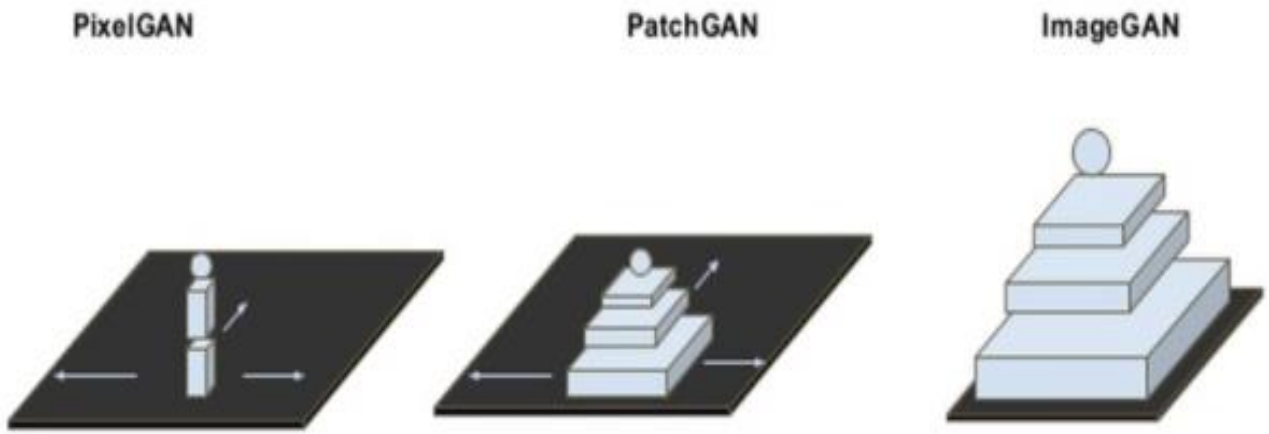
전체 이미지를 연산하여 판별하는 것 보다 연산의 수가 적고 빠름

G는 각각의 이미지 patch 조각의 Real/fake를 속이기 위해 학습과정이 진행

pix2pix architecture

-Discriminator





PixelGAN : 1x1

PatchGAN: NxN

ImageGAN: 전체

pix2pix architecture

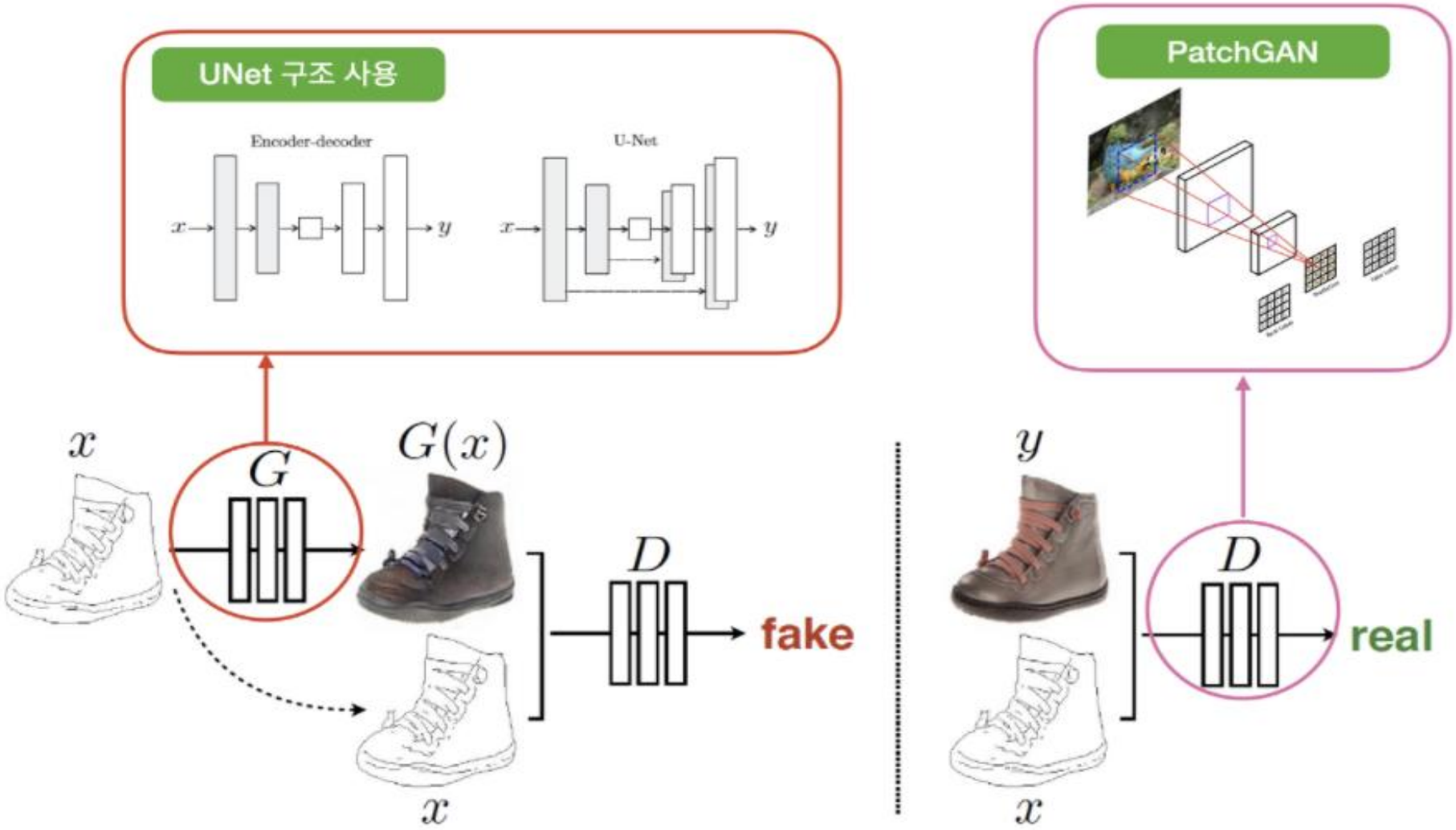
-Discriminator layers

70x70 discriminator architectures

C64 – C128 – C256 – C512

- 마지막 layer 이후 1d output return(sigmoid func)
- 첫번째 layer에서 BatchNorm 적용 x
- Leaky ReLU(slope = 0.2) 사용

pix2pix architecture



(그림 5) Pix2Pix Model Architecture.

pix2pix architecture

-Train / Optimization

<Train>
Jittering: 256x256 이미지는 286x286 크기로 resize된 후, random cropping을 통해
256x256으로 변환

모든 네트워크는 scratch로부터 학습

weight는 Gaussian dist(0, 0.02)를 따르는 랜덤 초기값을 가짐

<Optimization>
일반적인 GAN 접근법을 따름

D를 최적화하는 Objective를 2로 나누어 D가 G보다 상대적으로 빠르게 학습되지 않도록 함

minibatch SGD, Adam(lr = 0.0002, beta = (0.5, 0.999)) 사용

Results

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.

L1과 cGAN을 같이 사용했을 때 더 좋은 결과

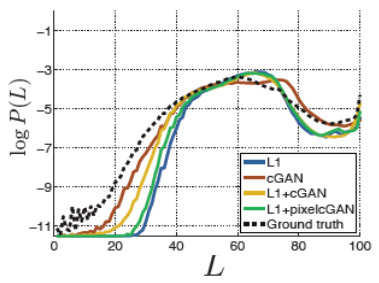
Results

Figure 5: Adding skip connections to an encoder-decoder to create a "U-Net" results in much higher quality results.

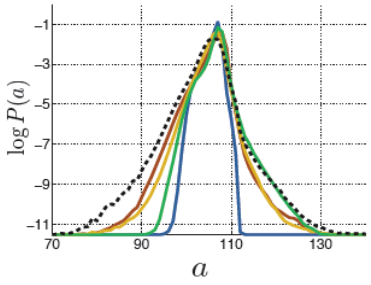
Encoder-decoder 보다 U-Net 구조가 더 선명한 이미지 생성

Results

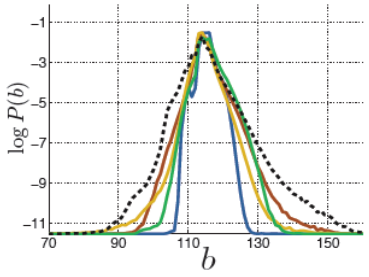
Figure 6: Patch size variations. Uncertainty in the output manifests itself differently for different loss functions. Uncertain regions become blurry and desaturated under L1. The 1x1 PixelGAN encourages greater color diversity but has no effect on spatial statistics. The 16x16 PatchGAN creates locally sharp results, but also leads to tiling artifacts beyond the scale it can observe. The 70×70 PatchGAN forces outputs that are sharp, even if incorrect, in both the spatial and spectral (colorfulness) dimensions. The full 286×286 ImageGAN produces results that are visually similar to the 70×70 PatchGAN, but somewhat lower quality according to our FCN-score metric (Table 3). Please see `https://phillipi.github.io/pix2pix/` for additional examples.



(a)



(b)



(c)

Loss	Histogram intersection against ground truth L	a	b
L1	0.81	0.69	0.70
cGAN	**0.87**	0.74	**0.84**
L1+cGAN	0.86	**0.84**	0.82
PixelGAN	0.83	0.68	0.78

(d)

Figure 7: Color distribution matching property of the cGAN, tested on Cityscapes. (c.f. Figure 1 of the original GAN paper [24]). Note that the histogram intersection scores are dominated by differences in the high probability region, which are imperceptible in the plots, which show log probability and therefore emphasize differences in the low probability regions.

Patch size에 따라 다른 결과

References

https://arxiv.org/abs/1611.07004

https://greeksharifa.github.io/generative%20model/2019/04/07/Pix2Pix/

https://medium.com/humanscape-tech/paper-review-pix2pix-20418569e0c1

https://github.com/phillipi/pix2pix