생성모델 (Generative Models)

적대적 생성 모델 - 게임이론

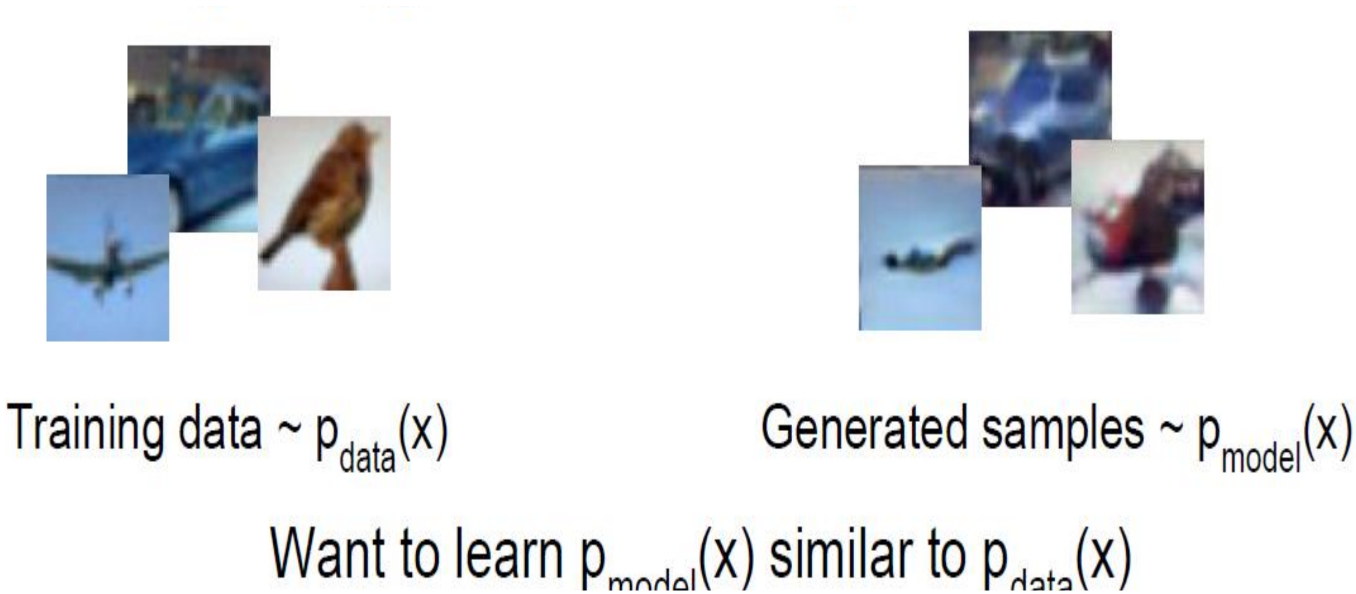
Dr. Rhee

Jyly 2018

생성모델 (Generative Models)

생성모델이란?

- 샘플 데이터가 주어졌을 때, 동일한 분포에서 새로운 샘플을 생성한다.



- Pmodel를 가지고 Pdata와 같은 분포의 새로운 데이터를 생성하고자 한다.

생성모델 분류와 계보

Taxonomy of Generative Models

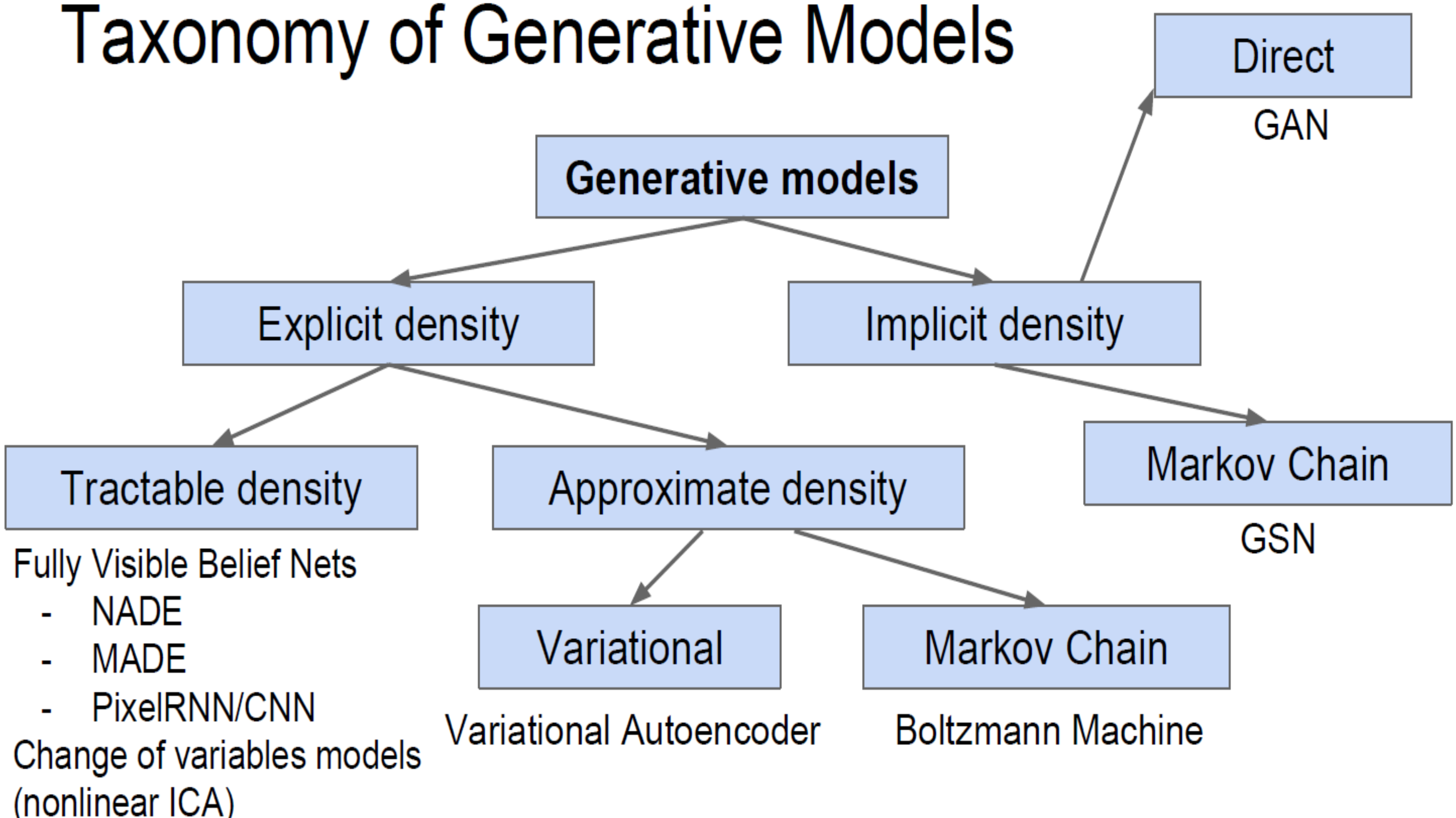


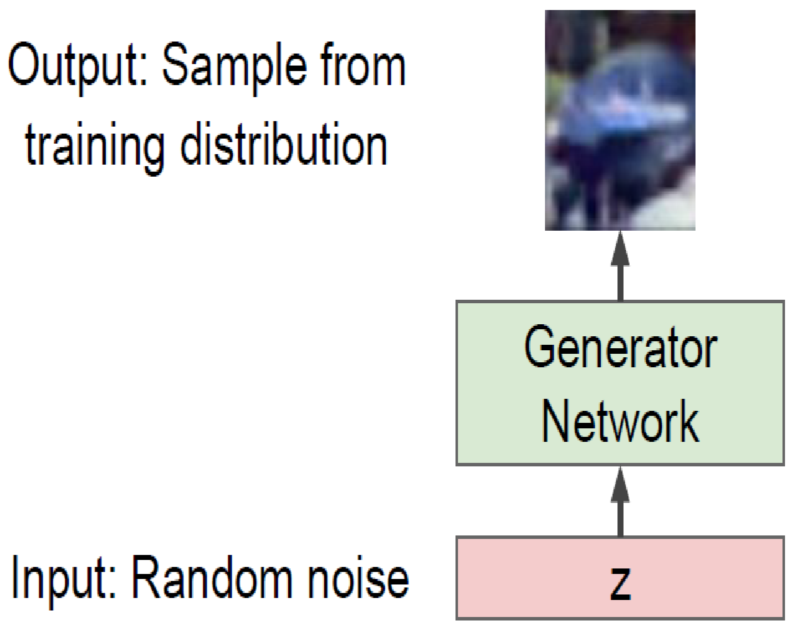
Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

적대적 생성모델 (GAN: Generative Adversarial Networks)

GAN 동기

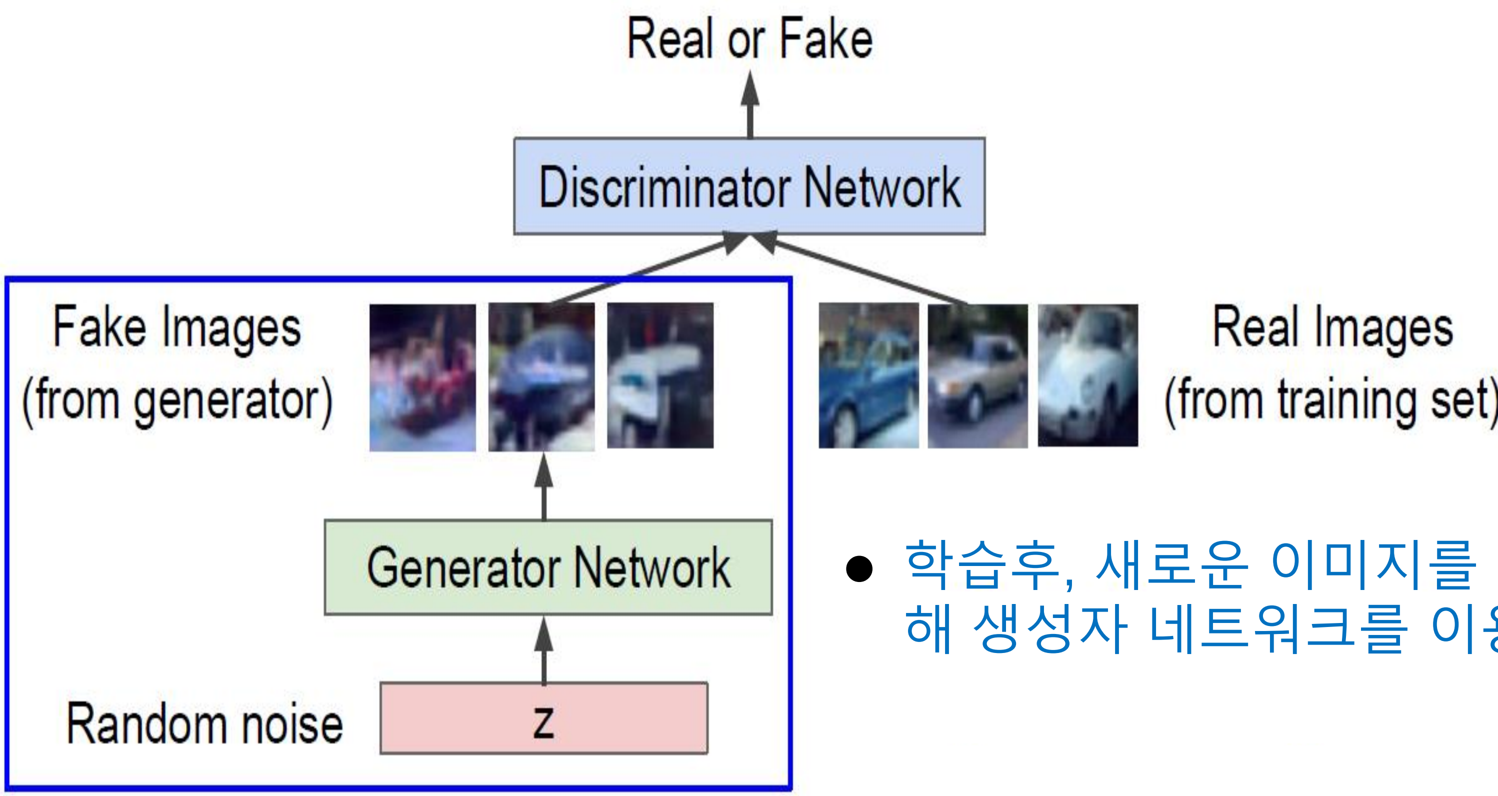
- 문제: 복잡하고, 고차원 훈련 데이터셋 분포로부터 샘플하기를 원한다.
- 해법: 단순한 분포(무작위 잡음) 로부터 샘플링해서훈련데이터셋 분포로 변환한다.

Q: 복잡한 변환을 표현하기 위해 무엇을 사용할까? (신경망)



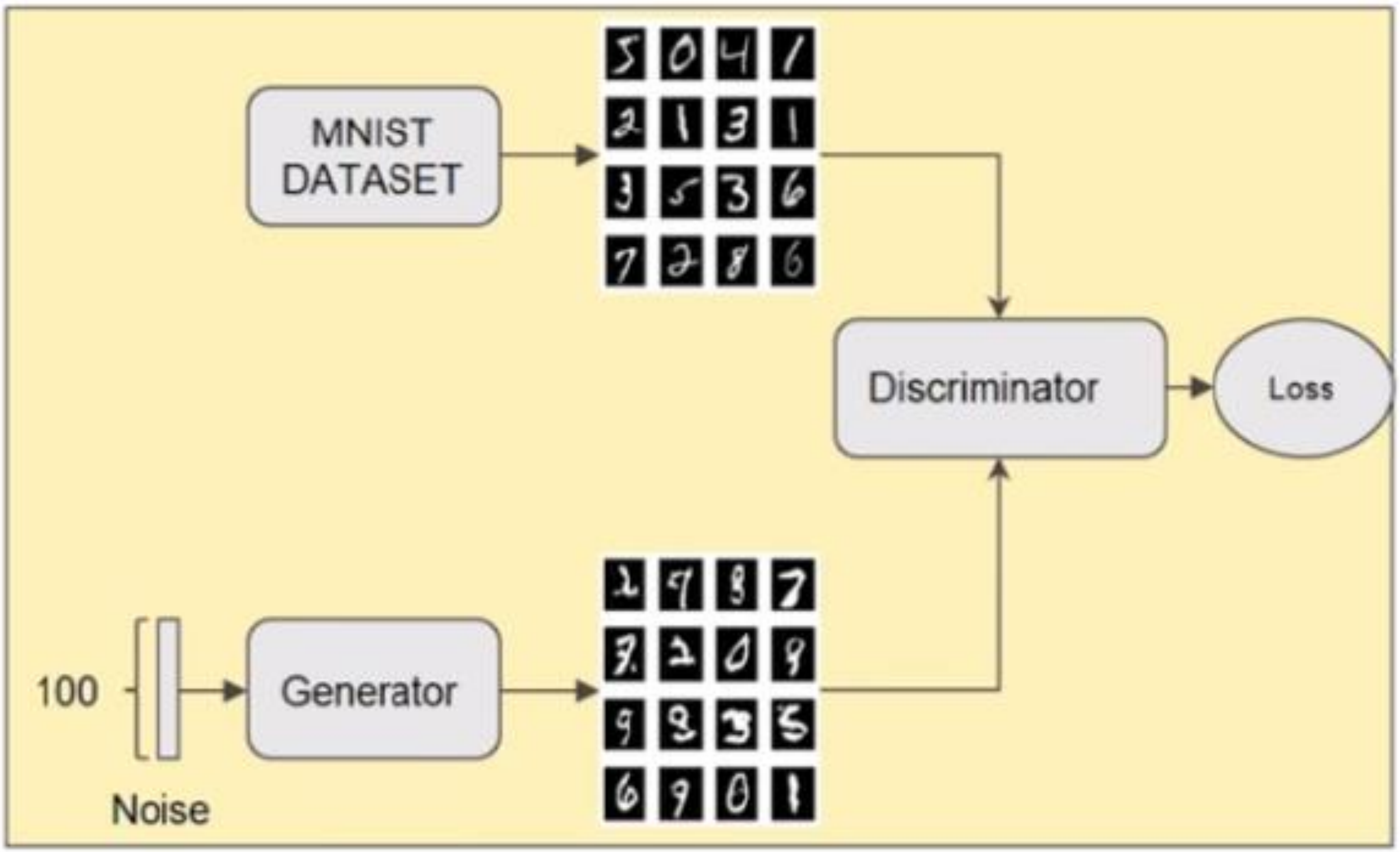
GAN의 훈련: 2인 선수 게임

- 생성자 네트워크: 진짜같이 보이는 이미지를 생성해서 판별자를 속이려고 한다.
- 판별자 네트워크: 진짜와 가짜 이미지를 구별하고자 한다.



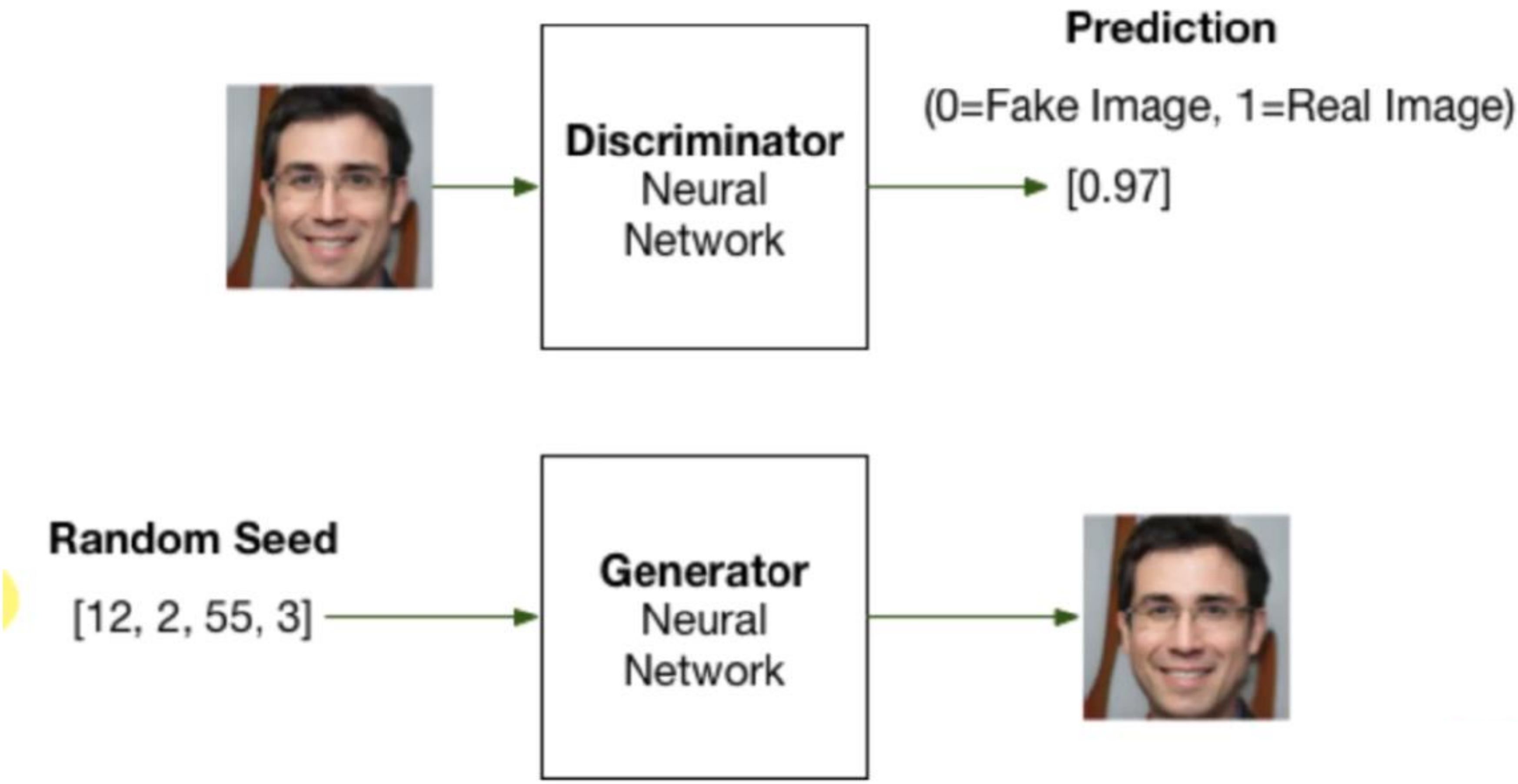
- 학습후, 새로운 이미지를 생성하기 위해 생성자 네트워크를 이용한다.

DCGAN - MNIST



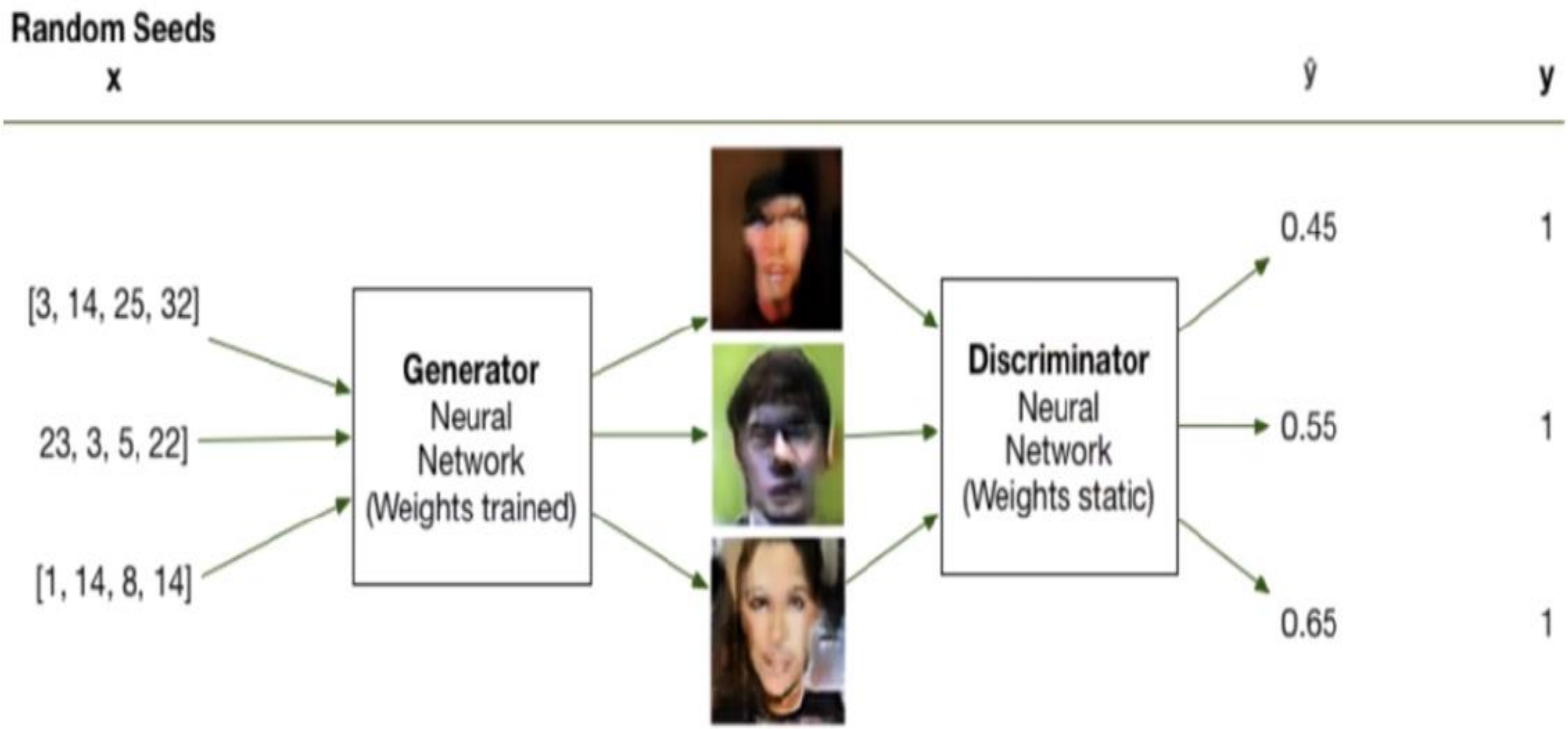
DCGAN – 얼굴 생성하기

- GAN 구조



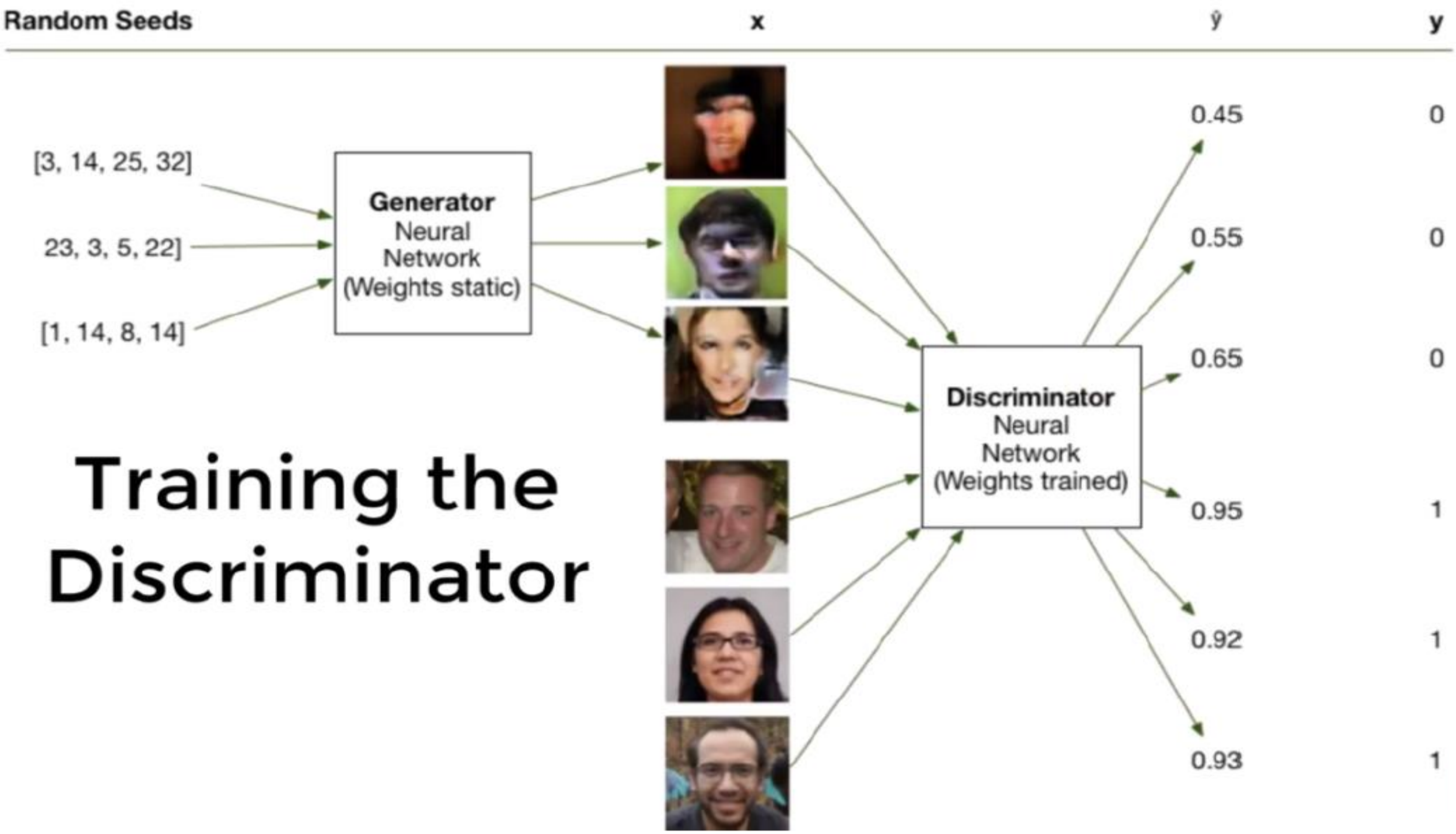
DCGAN – 얼굴 생성하기

- 생성자 신경망의 학습



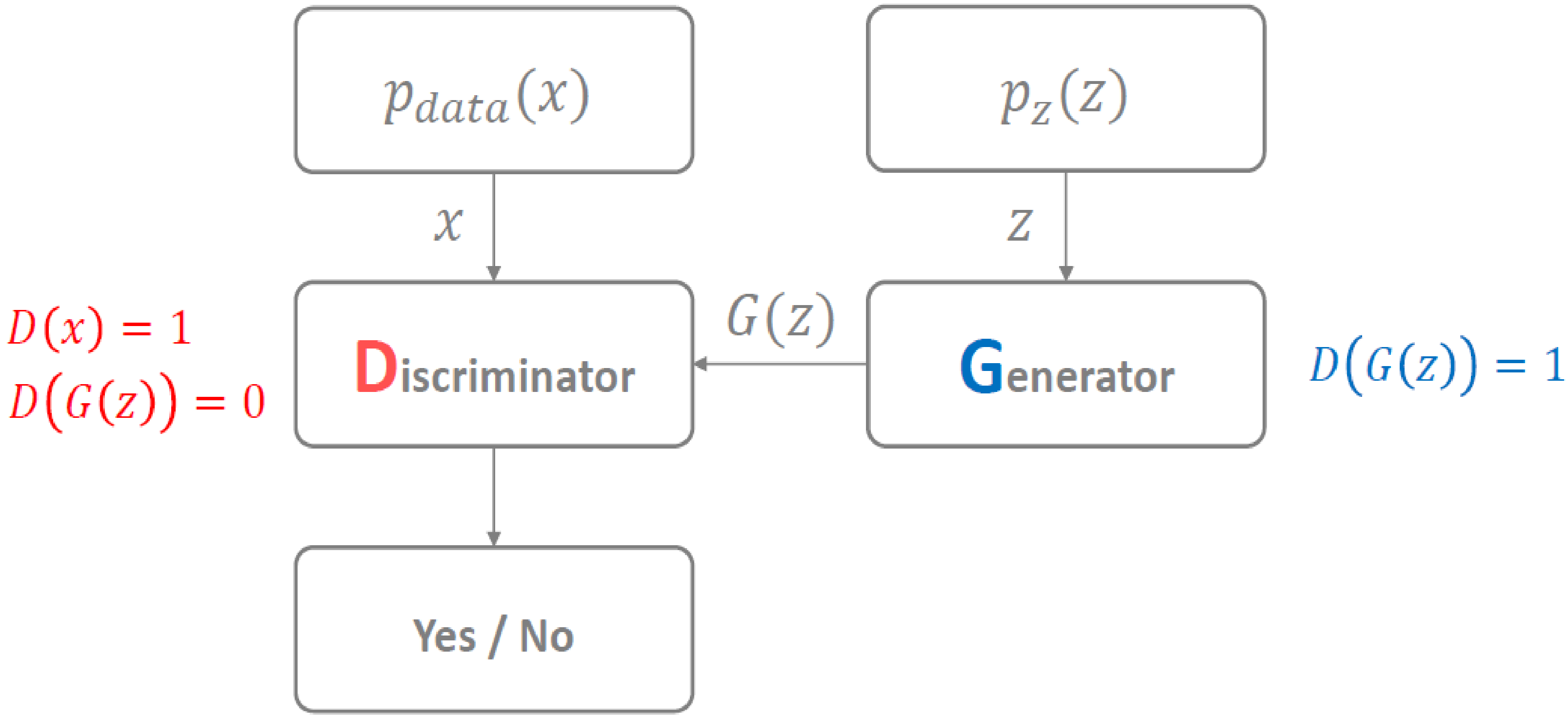
DCGAN – 얼굴 생성하기

- 판별자 신경망의 학습



Generative Adversarial Network(GAN)

- D와 G



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

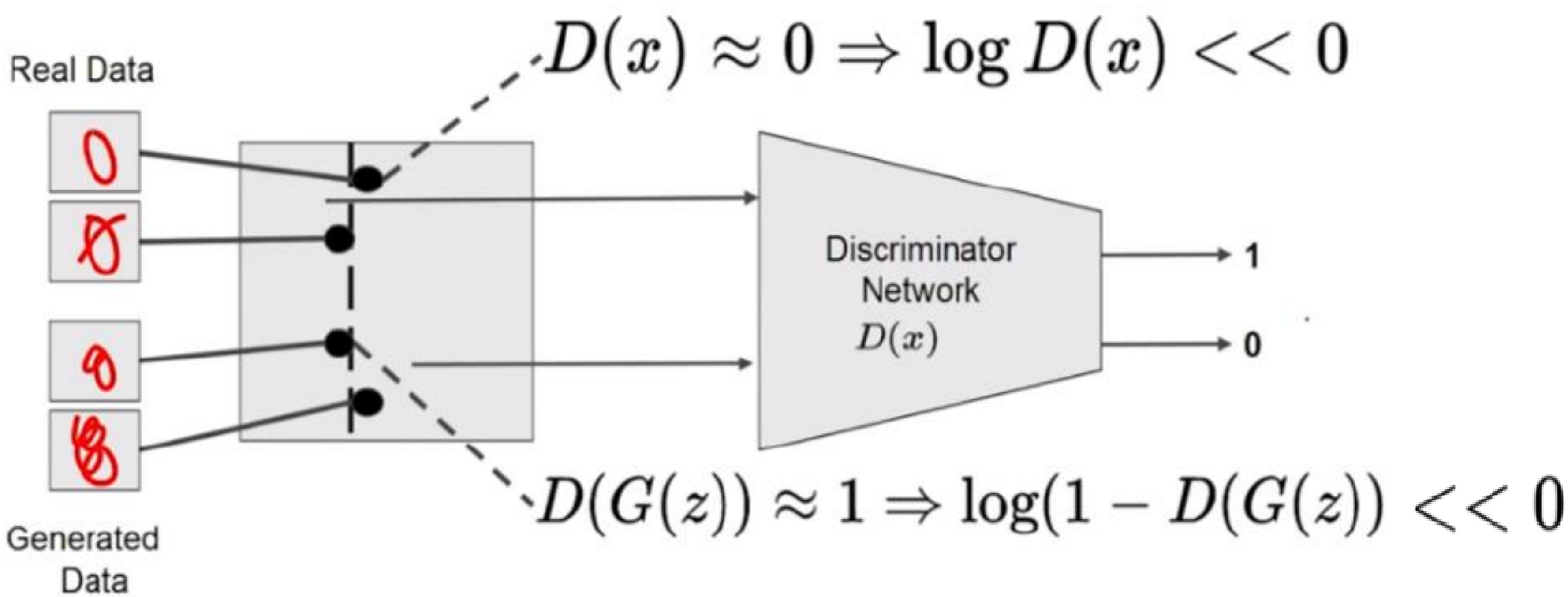
$$D^*, G^* = \min_G \max_D V(D,G)$$ GAN은 $G(z) \sim p_{data}(x)$로 만드는 것이 목적이다

Generative Adversarial Network(GAN)

- 판별자 네트워크의 학습 - 학습이전

$$\max_D \left[\mathbb{E}_{x\sim P_{data}(x)} \boxed{log(D(x))} + \mathbb{E}_{z\sim P_z(z)} \boxed{log(1-D(G(z)))} \right]$$

$D(x)$ should be 1 $\qquad$ $D(G(z))$ should be 0



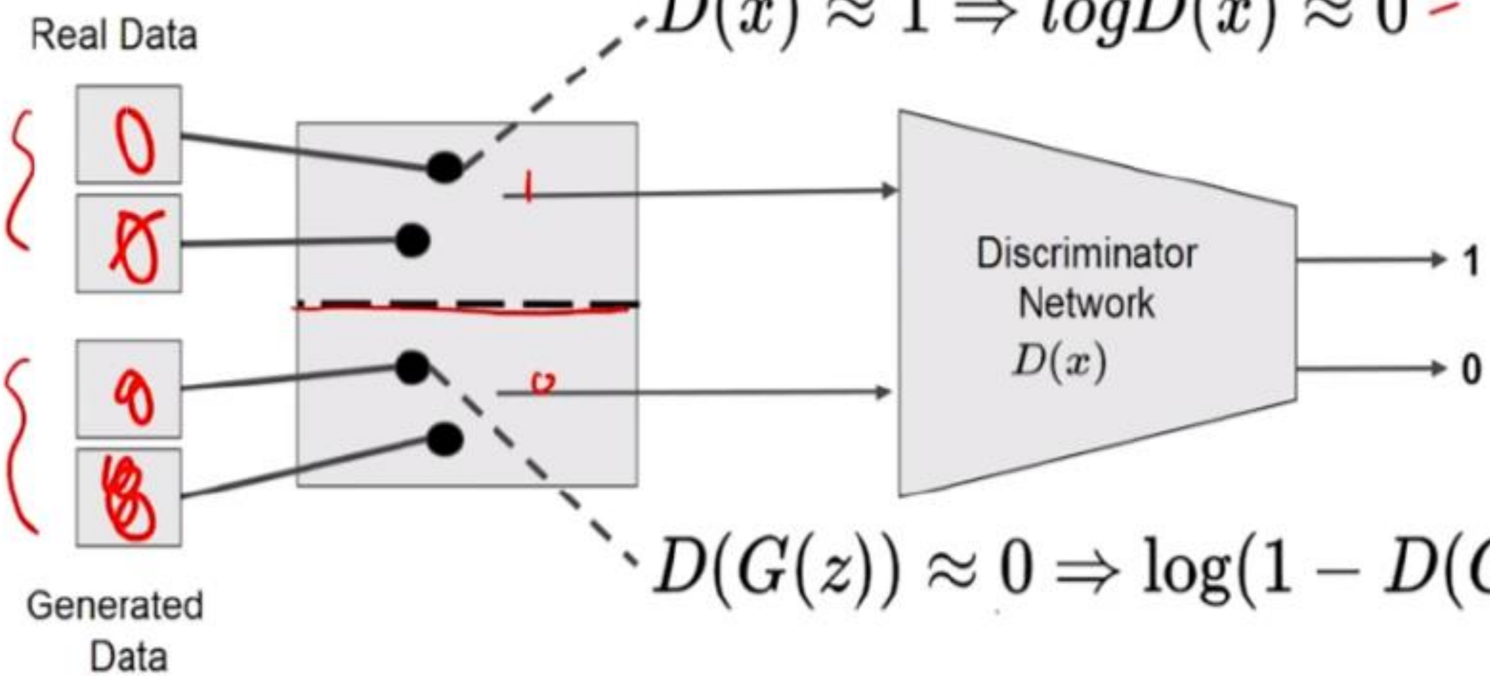
Generative Adversarial Network(GAN)

- 판별자 네트워크의 학습 – 학습 후

$$\max_D \left[\mathbb{E}_{x\sim P_{data}(x)} \boxed{log(D(x))} + \mathbb{E}_{z\sim P_z(z)} \boxed{log(1 - D(G(z)))}\right]$$

$D(x)$ should be 1 $\qquad$ $D(G(z))$ should be 0

$$D(x) \approx 1 \Rightarrow logD(x) \approx 0$$



$$D(G(z)) \approx 0 \Rightarrow \log(1 - D(G(z)) \approx 0$$

Generative Adversarial Network(GAN)

- 생성자 네트워크의 학습

$$\min_G [\boxed{\mathbb{E}_{x\sim P_{data}} (\log D(x))} + \mathbb{E}_{z\sim P(z)} (\log(1 - D(G(z))))]$$

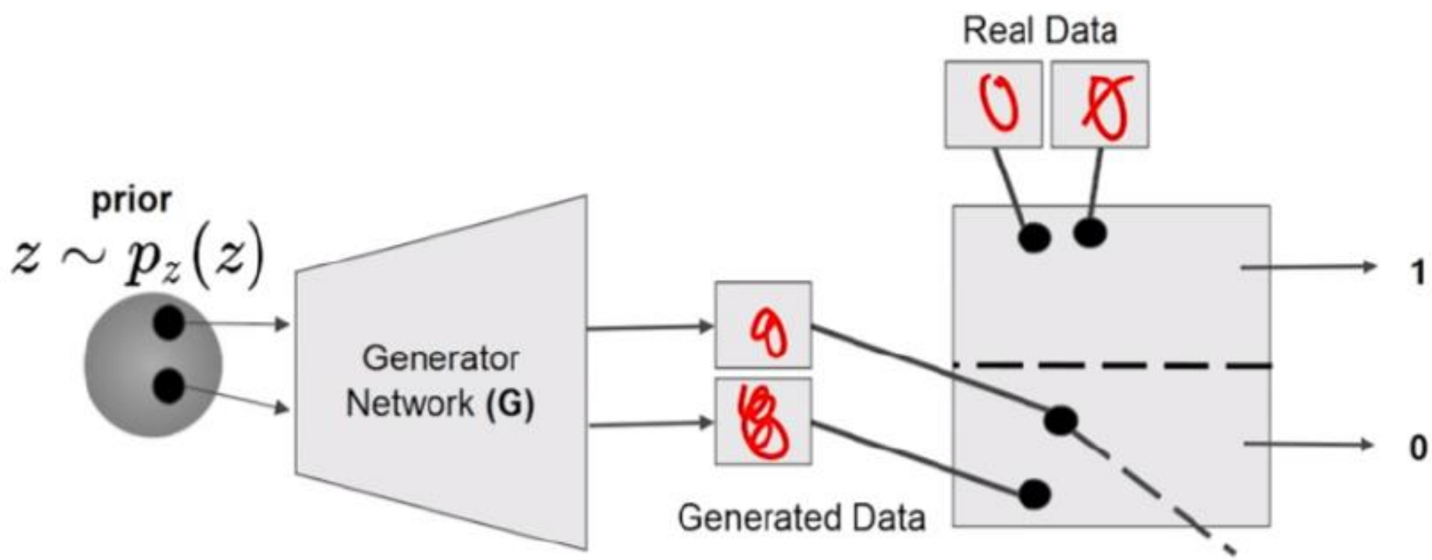
$$\Rightarrow \min_G [\mathbb{E}_{z\sim P(z)} \boxed{(\log(1 - D(G(z))))}]$$

Generative Adversarial Network(GAN)

- 생성자 네트워크 학습 – 학습 이전

$$\max_G \left[\mathbb{E}_{z\sim P_z(z)} \boxed{log(D(G(z)))}\right]$$

$D(G(z))$ should be **1**



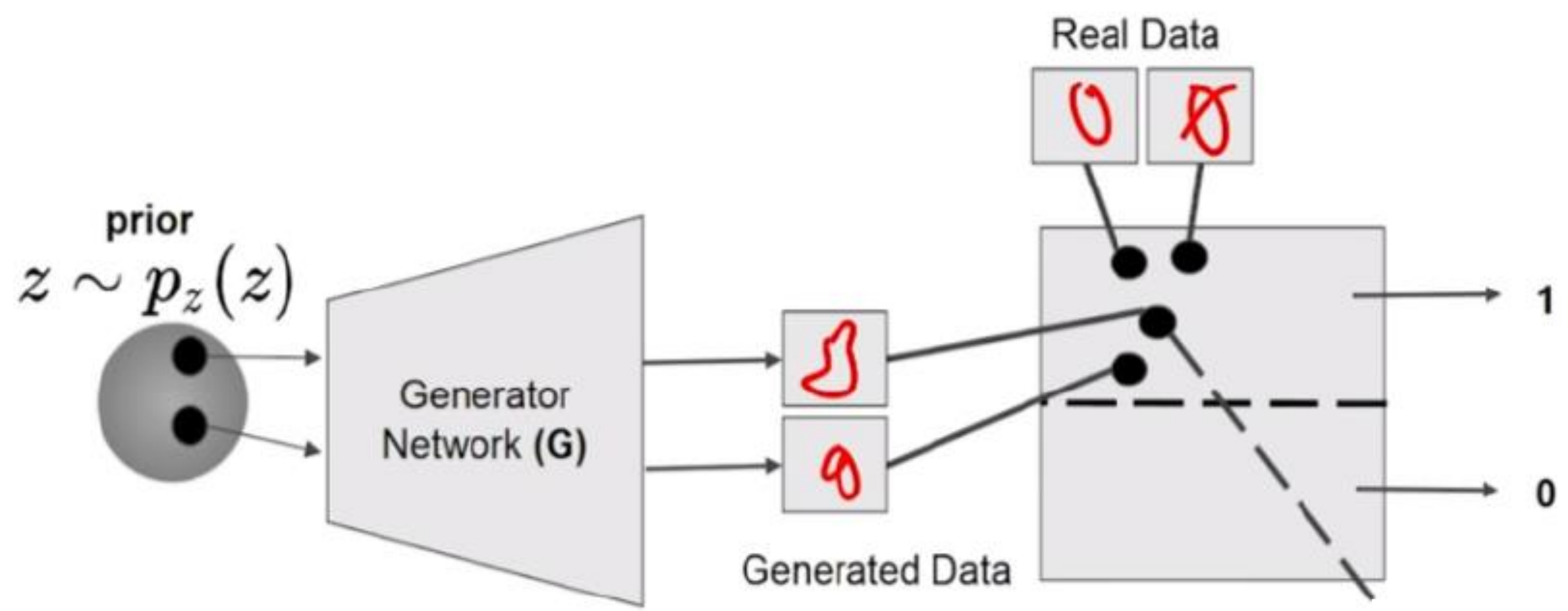
$$D(G(z)) \approx 0 \Rightarrow \log(D(G(z)) << 0$$

Generative Adversarial Network(GAN)

- 생성자 네트워크 학습 – 학습 후

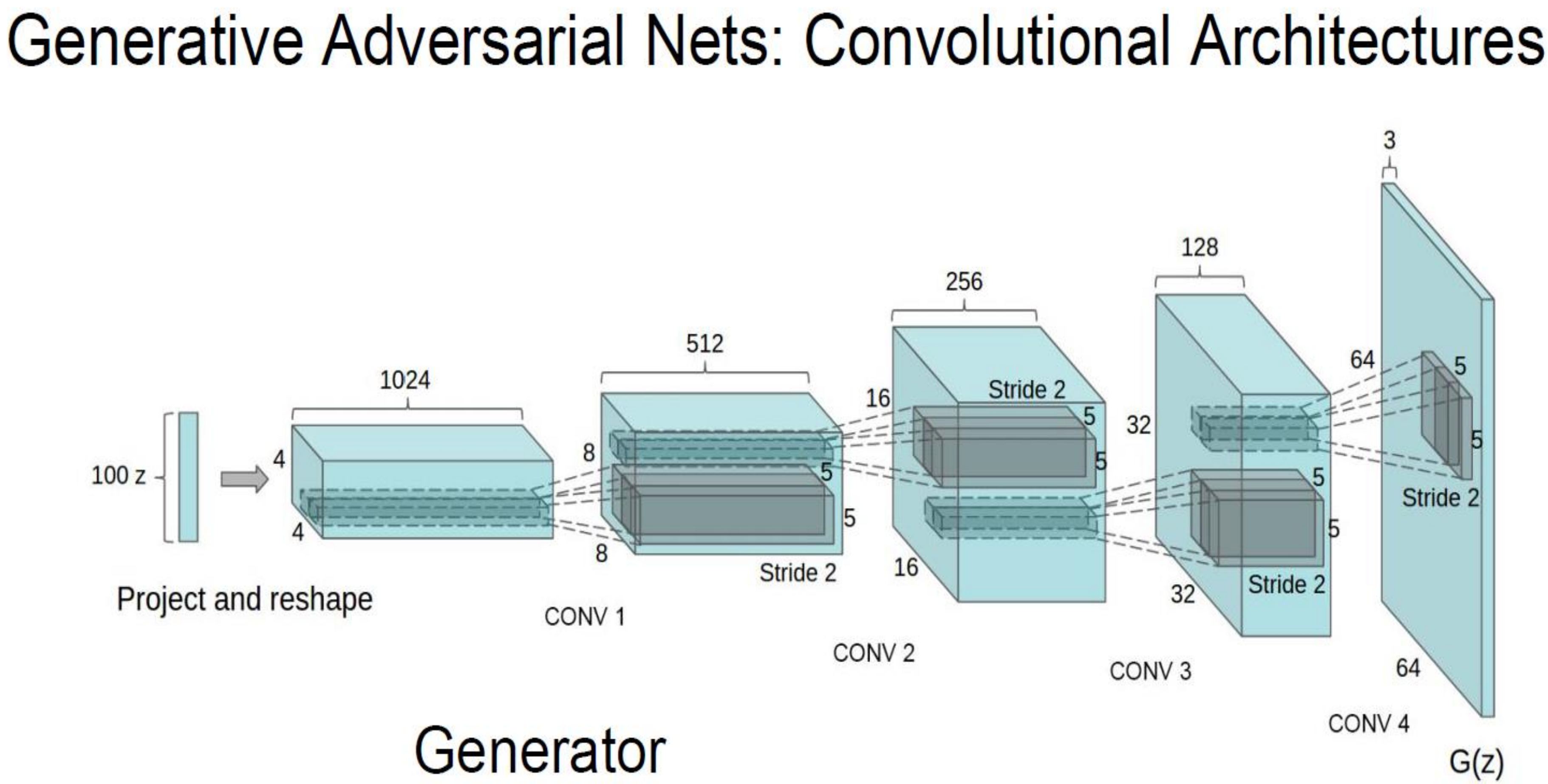
$$\max_G \left[\mathbb{E}_{z\sim P_z(z)} \boxed{log(D(G(z)))}\right]$$

$D(G(z))$ should be **1**



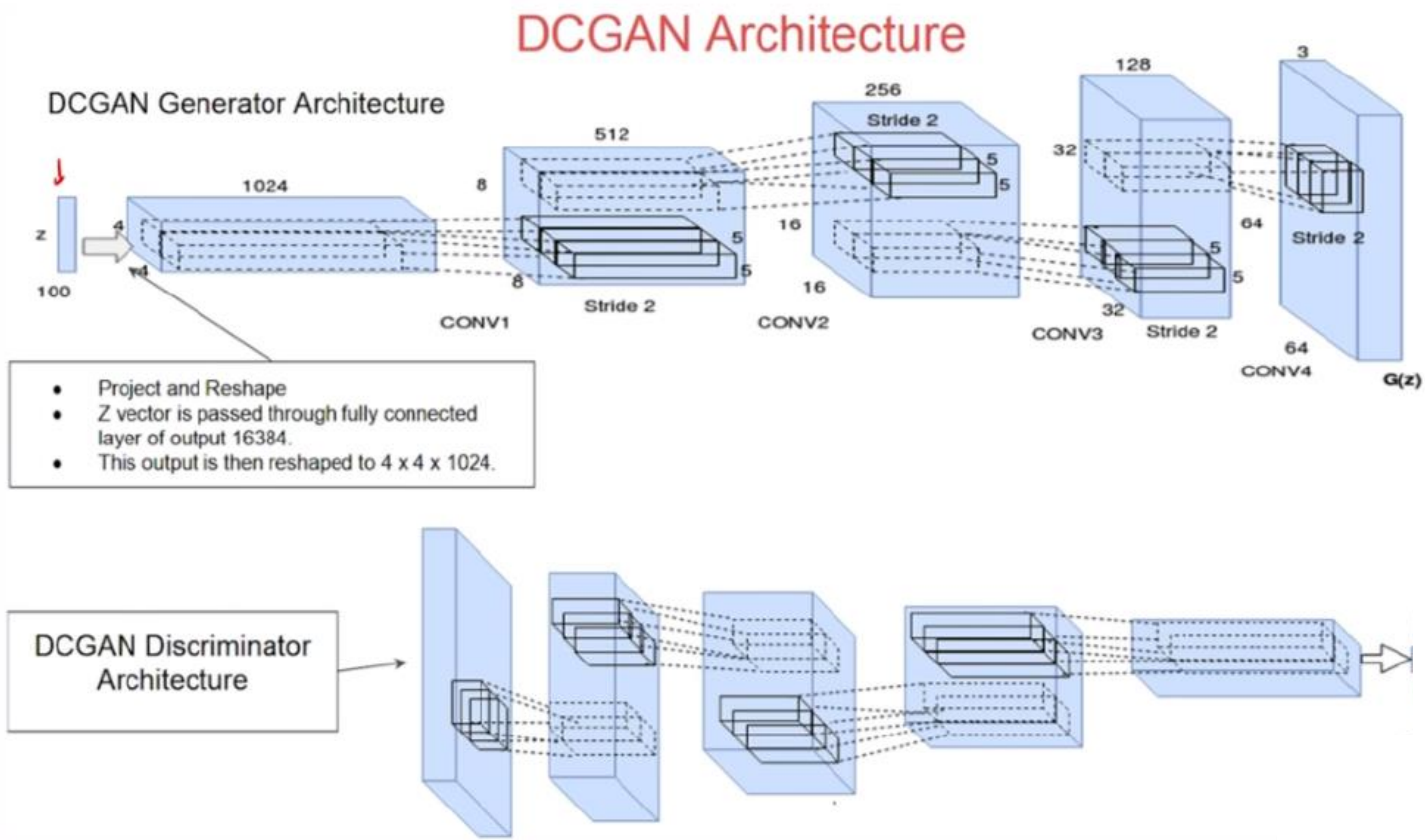
$$D(G(z)) \approx 1 \Rightarrow \log(D(G(z))) \approx 0$$

DCGAN (Deep Convolutional GAN)



Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016

DCGAN (Deep Convolutional GAN) 전체 구조



DCGAN – 얼굴 생성하기

- 손실함수의 정의

```
# This method returns a helper function to compute cross entropy loss
cross_entropy = tf.keras.losses.BinaryCrossentropy(from_logits=True)

def discriminator_loss(real_output, fake_output):
    real_loss = cross_entropy(tf.ones_like(real_output), real_output)
    fake_loss = cross_entropy(tf.zeros_like(fake_output), fake_output)
    total_loss = real_loss + fake_loss
    return total_loss

def generator_loss(fake_output):
    return cross_entropy(tf.ones_like(fake_output), fake_output)
```

```
generator_optimizer = tf.keras.optimizers.Adam(1.5e-4,0.5)
discriminator_optimizer = tf.keras.optimizers.Adam(1.5e-4,0.5)
```

DCGAN – 얼굴 생성하기

- 그래디언트 스텝을 통한 최적화

```
# Notice the use of `tf.function`
# This annotation causes the function to be "compiled".
@tf.function
def train_step(images):
  seed = tf.random.normal([BATCH_SIZE, SEED_SIZE])

  with tf.GradientTape() as gen_tape, tf.GradientTape() as disc_tape:
    generated_images = generator(seed, training=True)

    real_output = discriminator(images, training=True)
    fake_output = discriminator(generated_images, training=True)

    gen_loss = generator_loss(fake_output)
    disc_loss = discriminator_loss(real_output, fake_output)


    gradients_of_generator = gen_tape.gradient(gen_loss, generator.trainable_variable
s)
    gradients_of_discriminator = disc_tape.gradient(disc_loss, discriminator.trainabl
e_variables)

    generator_optimizer.apply_gradients(zip(gradients_of_generator, generator.trainab
le_variables))
    discriminator_optimizer.apply_gradients(zip(gradients_of_discriminator, discrimin
ator.trainable_variables))
  return gen_loss,disc_loss
```

DCGAN – 얼굴 생성하기

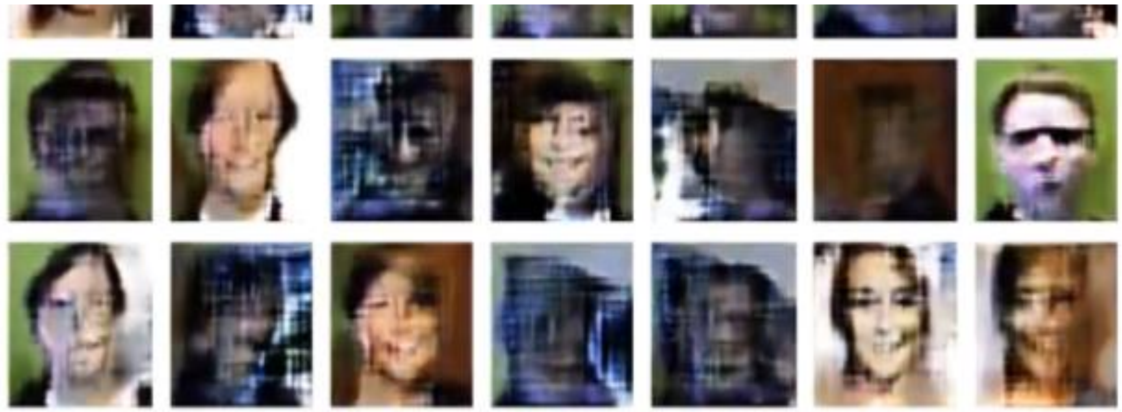
- GAN의 학습

```
def train(dataset, epochs):
  fixed_seed = np.random.normal(0, 1, (PREVIEW_ROWS * PREVIEW_COLS, SEED_SIZE))
  start = time.time()

  for epoch in range(epochs):
    epoch_start = time.time()

    gen_loss_list = []
    disc_loss_list = []

    for image_batch in dataset:
      t = train_step(image_batch)
      gen_loss_list.append(t[0])
      disc_loss_list.append(t[1])

    g_loss = sum(gen_loss_list) / len(gen_loss_list)
    d_loss = sum(disc_loss_list) / len(disc_loss_list)

    epoch_elapsed = time.time()-epoch_start
    print (f'Epoch {epoch+1}, gen loss={g_loss},disc loss={d_loss}, {hms_string(epoch
_elapsed)}')
    save_images(epoch,fixed_seed)

  elapsed = time.time()-start
  print (f'Training time: {hms_string(elapsed)}')
```

DCGAN – 얼굴 생성하기

- GNN의 학습

```
train(train_dataset, EPOCHS)
```

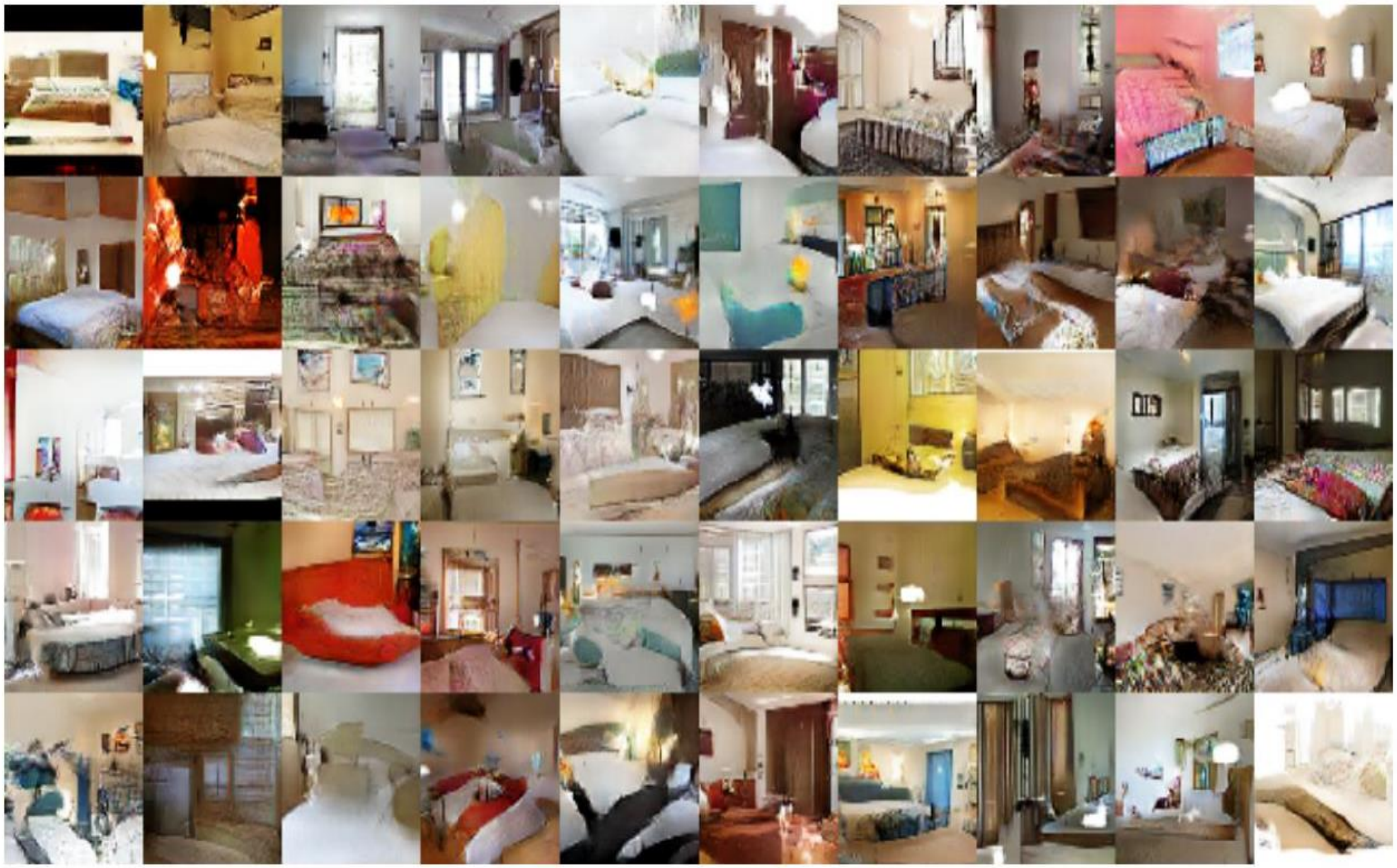


```
Epoch 1, gen loss=0.6737478375434875,
Epoch 2, gen loss=0.6754337549209595,
Epoch 3, gen loss=0.6792119741439819,
Epoch 4, gen loss=0.6704637408256531,
Epoch 5, gen loss=0.6726831197738647,
Epoch 6, gen loss=0.6760282516479492,
Epoch 7, gen loss=0.6668657660484314,
Epoch 8, gen loss=0.6794514060020447,
Epoch 9, gen loss=0.667030930519104,disc loss=1.099564552307129, 0:01:26.83
Epoch 10, gen loss=0.6715224385261536,disc loss=1.0793499946594238, 0:01:26.66
Epoch 11, gen loss=0.6730715036392212,disc loss=1.078121304512024, 0:01:26.76
Epoch 12, gen loss=0.6647288203239441,disc loss=1.1045489311218262, 0:01:26.72
Epoch 13, gen loss=0.6678280234336853,disc loss=1.0965650081634521, 0:01:26.85
Epoch 14, gen loss=0.6659991145133972,disc loss=1.0995577573776245, 0:01:26.63
Epoch 15, gen loss=0.6716247797012329,disc loss=1.0771788358688354, 0:01:26.71
Epoch 16, gen loss=0.6729282140731812,disc loss=1.070194125175476, 0:01:26.70
Epoch 17, gen loss=0.5819647312164307,disc loss=1.2202180624008179, 0:01:26.77
Epoch 18, gen loss=0.6757639646530151,disc loss=1.065081238746643, 0:01:26.61
```

GAN

Generative Adversarial Nets: Convolutional Architectures

Samples from the model look amazing!

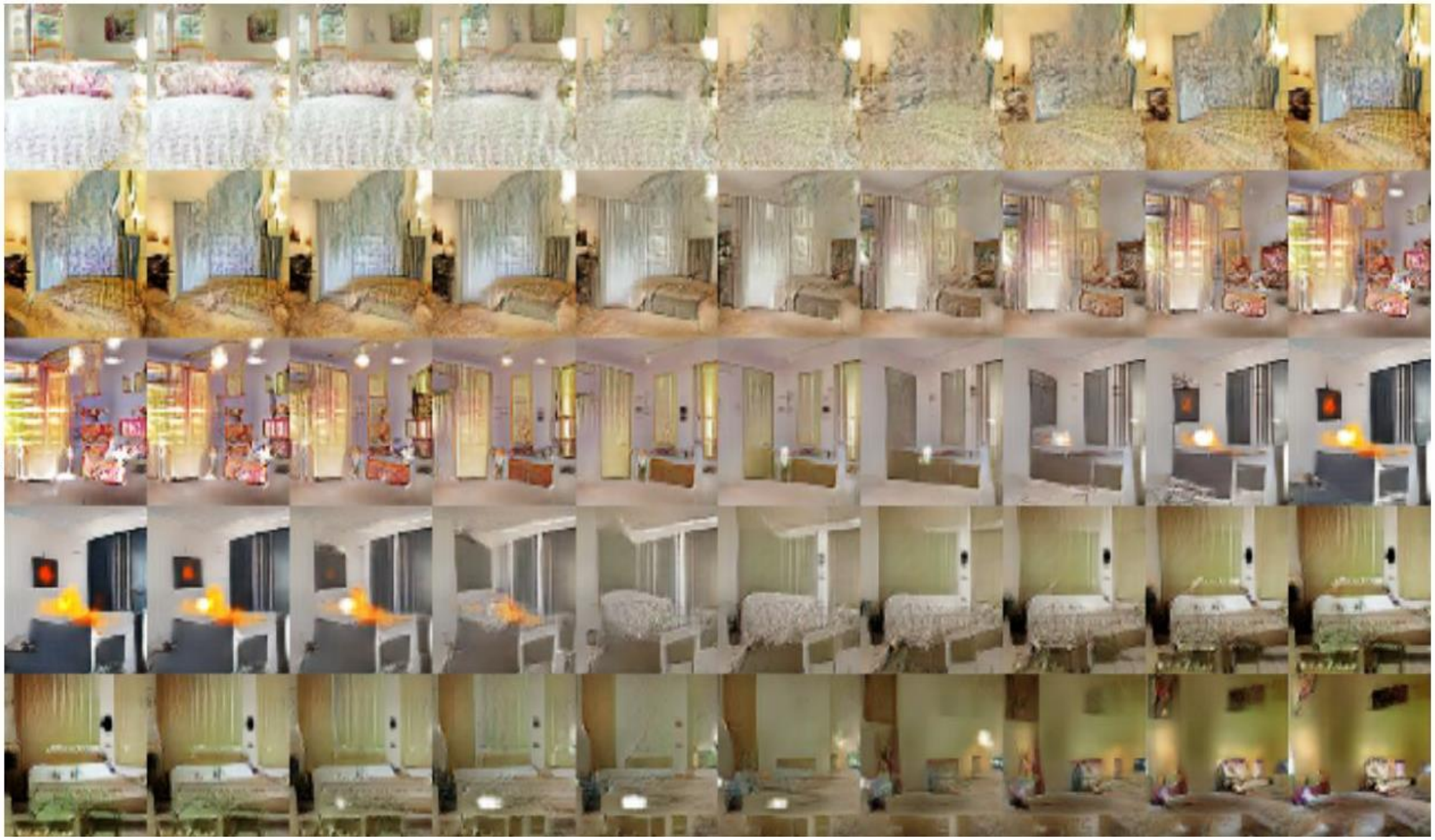


Radford et al, ICLR 2016

GAN

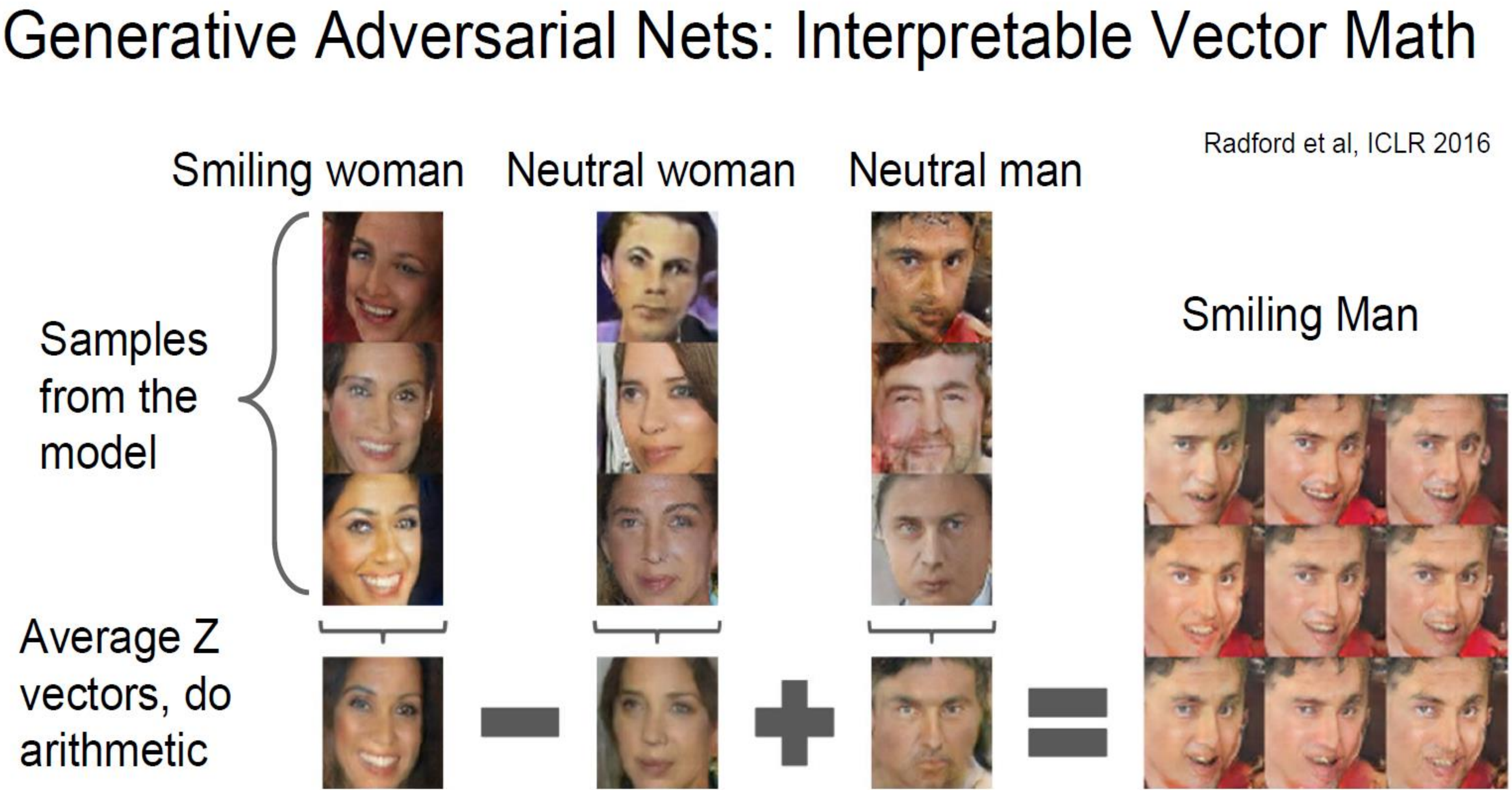
Generative Adversarial Nets: Convolutional Architectures

Interpolating between random points in laten space

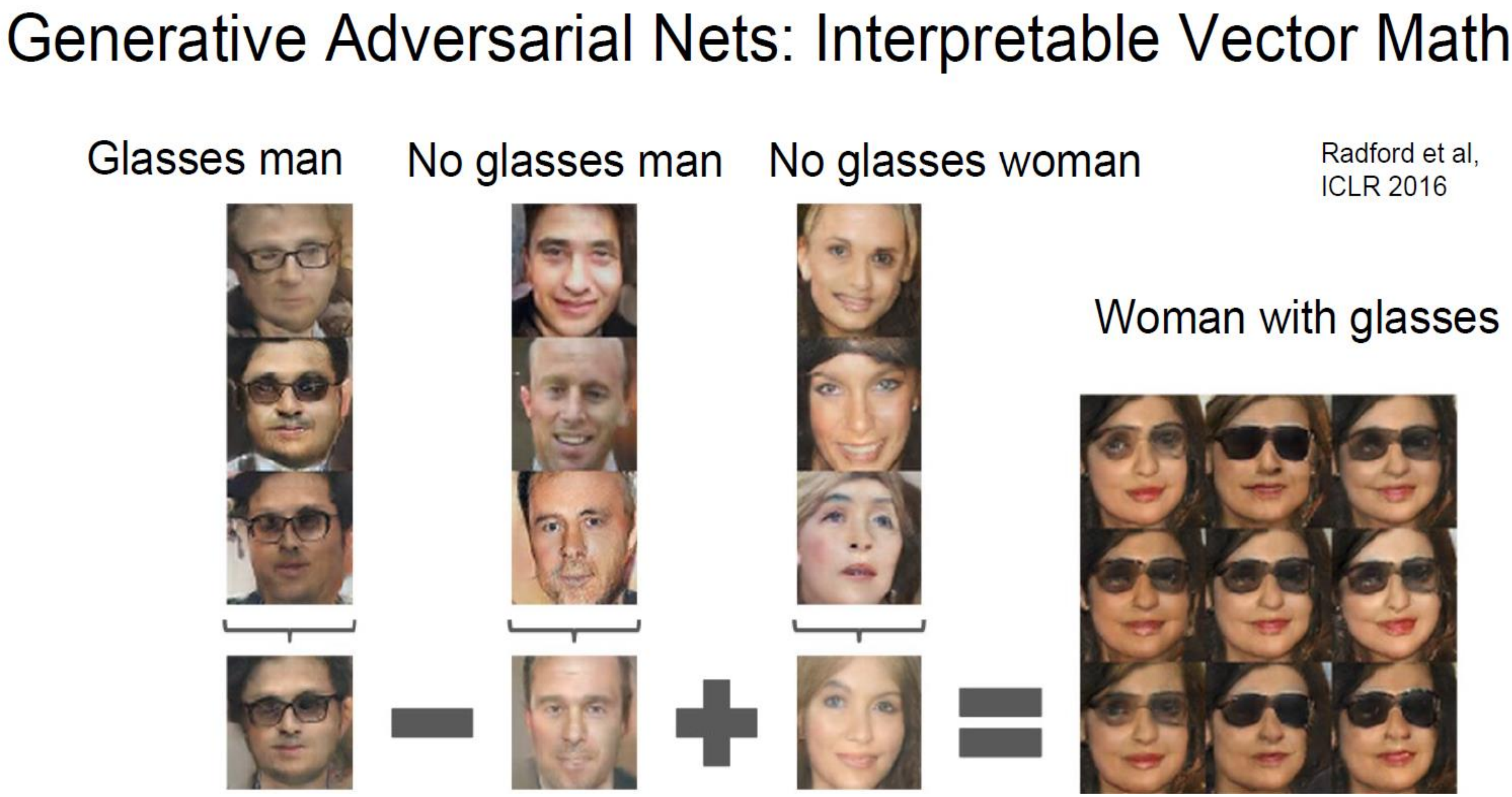


Radford et al,
ICLR 2016

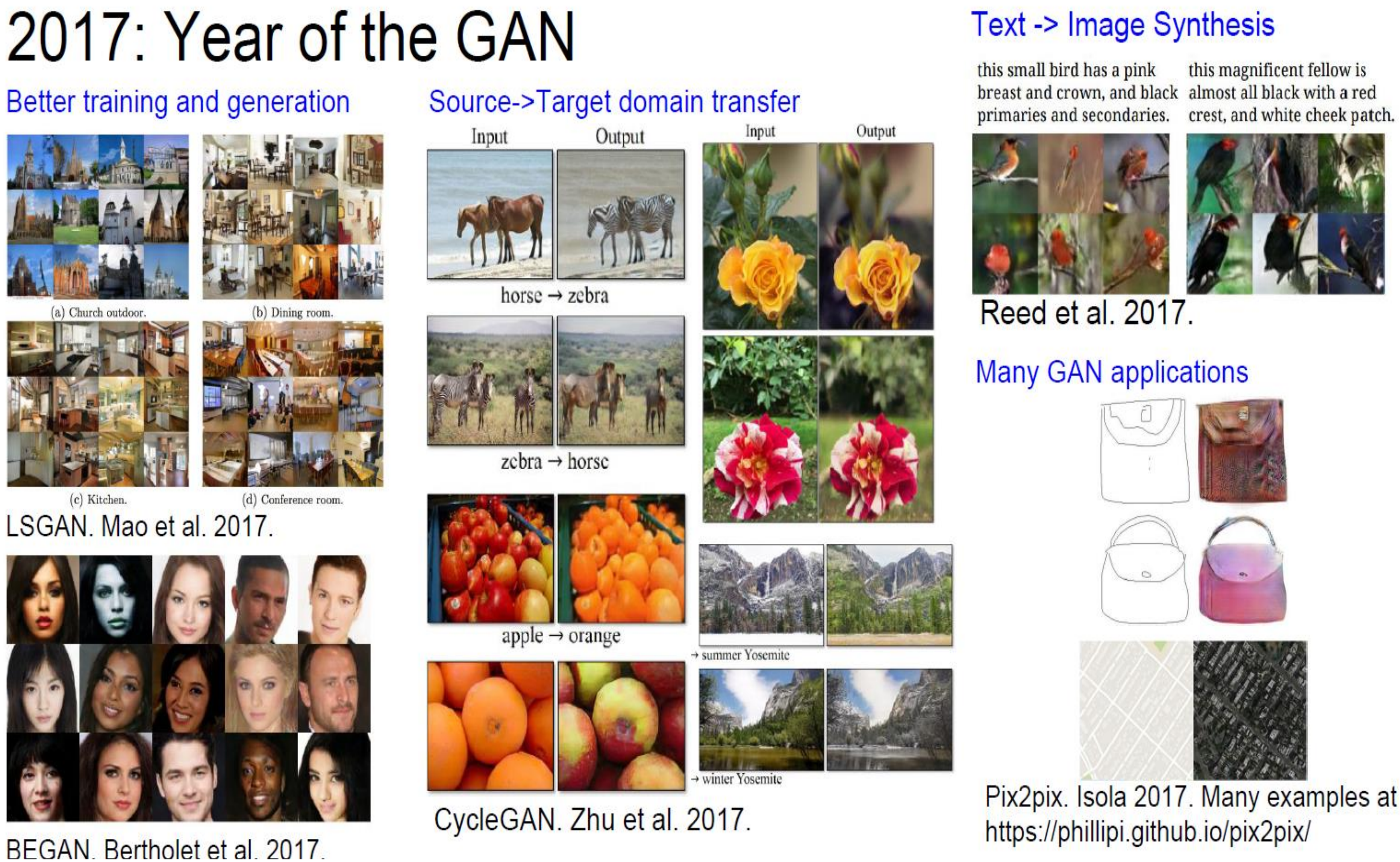
GAN


Generative Adversarial Nets: Interpretable Vector Math
Radford et al, ICLR 2016
Smiling woman
Neutral woman
Neutral man
Smiling Man
Samples from the model
Average Z vectors, do arithmetic

GAN


Generative Adversarial Nets: Interpretable Vector Math
Glasses man
No glasses man
No glasses woman
Radford et al,
ICLR 2016
Woman with glasses

GAN


2017: Year of the GAN
Better training and generation
(a) Church outdoor.
(b) Dining room.
(c) Kitchen.
(d) Conference room.
LSGAN. Mao et al. 2017.
BEGAN. Bertholet et al. 2017.
Source->Target domain transfer
Input
Output
horse → zebra
zebra → horse
apple → orange
Input
Output
→ summer Yosemite
→ winter Yosemite
CycleGAN. Zhu et al. 2017.
Text -> Image Synthesis
this small bird has a pink breast and crown, and black primaries and secondaries.
this magnificent fellow is almost all black with a red crest, and white cheek patch.
Reed et al. 2017.
Many GAN applications
Pix2pix. Isola 2017. Many examples at https://phillipi.github.io/pix2pix/

GAN

"The GAN Zoo"

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

https://github.com/hindupuravinash/the-gan-zoo

GAN의 응용

Conditional GAN

- Conditional GAN

(a) A set of human models $\{x_i\}_{i=1}^{16}$.

(b) A set of articles $\{y_j\}_{j=1}^{16}$, not present on the images in (a).

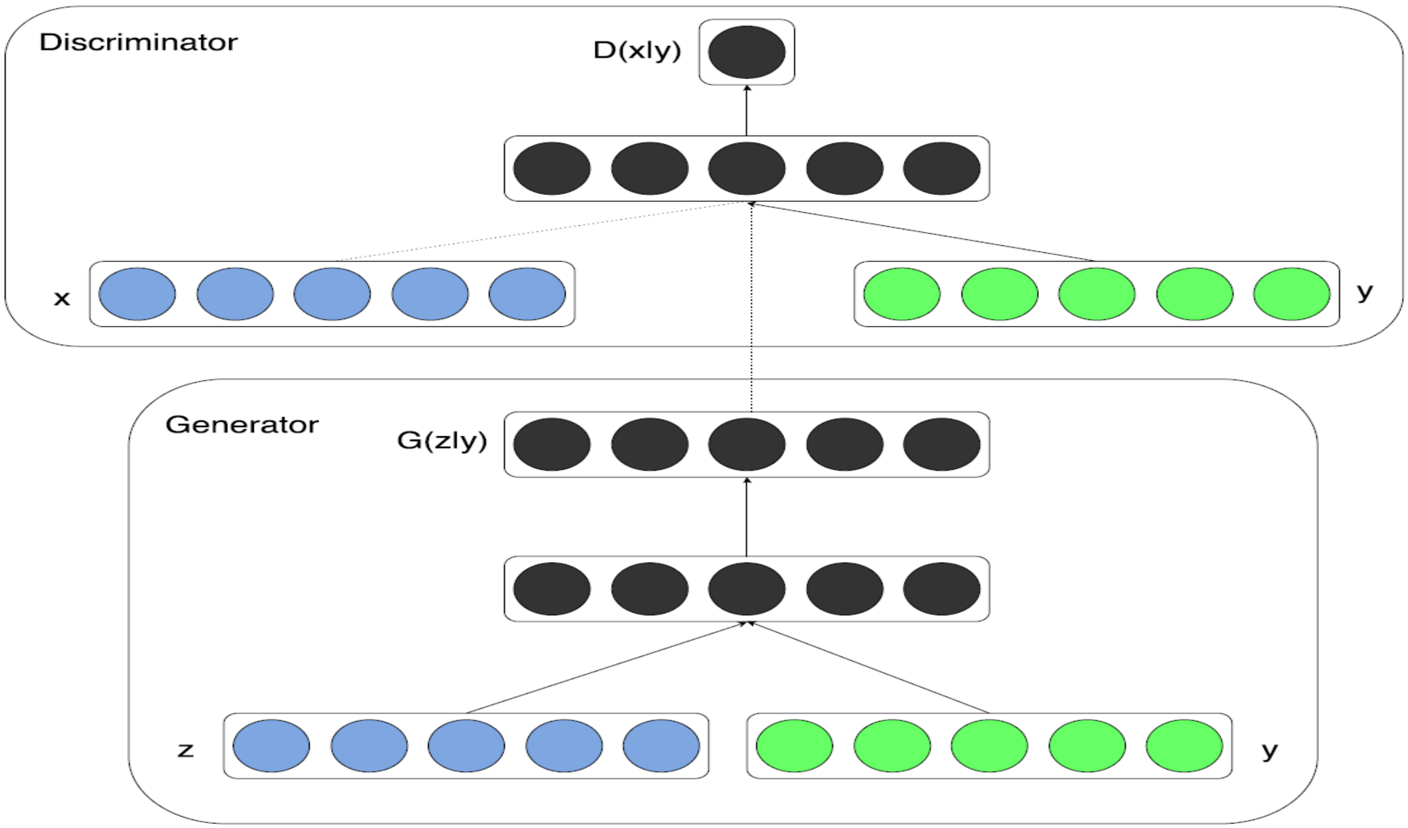
(c) $\{x_1^j\}_{j=1}^{16}$ generated from a fixed human image x_1 and the article set $\{y_i\}_{i=1}^{16}$. The image x_1 is in the top left corner of (a).

(d) $\{x_i^1\}_{i=1}^{16}$ generated from a fixed article image y_1 and different people images $\{x_i\}_{i=1}^{16}$. The article image y_1 is in the top left corner of (b).

http://openaccess.thecvf.com/content_ICCV_2017_workshops/w32/html/Jetchev_The_Conditional_Analogy_ICCV_2017_paper.html

Conditional GAN

- Conditional GAN

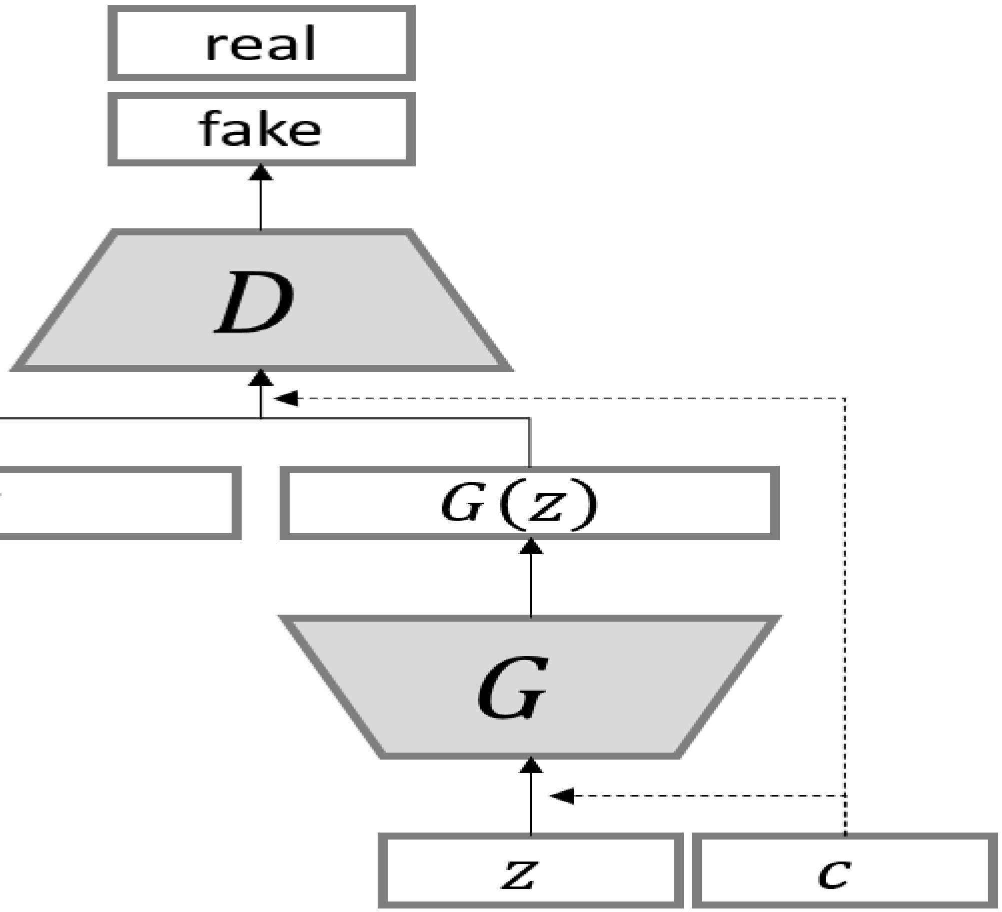


http://openaccess.thecvf.com/content_ICCV_2017_workshops/w32/html/Jetchev_The_Conditional_Analogy_ICCV_2017_paper.html

Conditional GAN

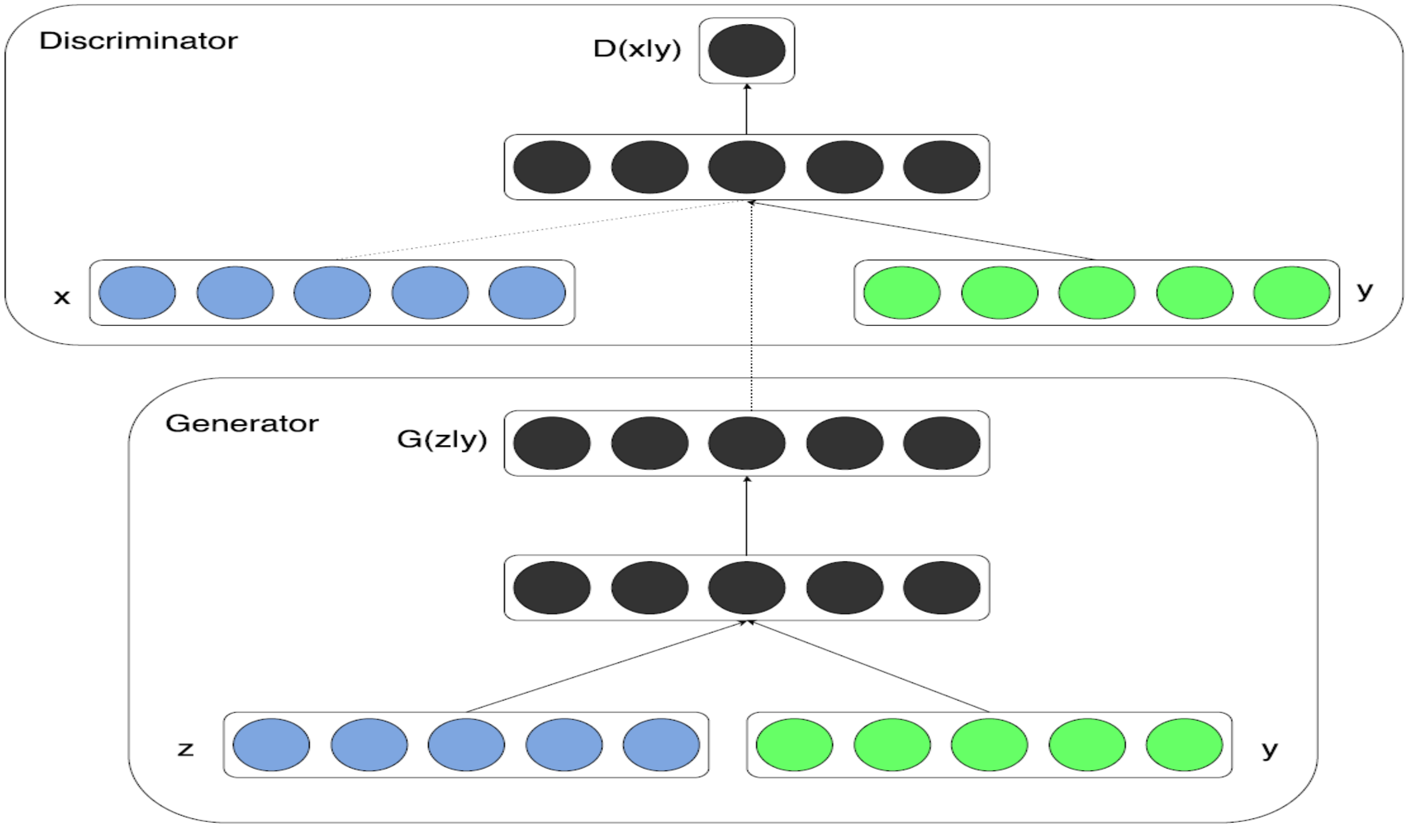
- Conditional GAN

CGAN



Conditional GAN

- Conditional GAN



http://openaccess.thecvf.com/content_ICCV_2017_workshops/w32/html/Jetchev_The_Conditional_Analogy_ICCV_2017_paper.html

Conditional GAN을 활용한 이미지 변환

- Visual Tranformation

A (input) A' (output) B (output) B' (input)

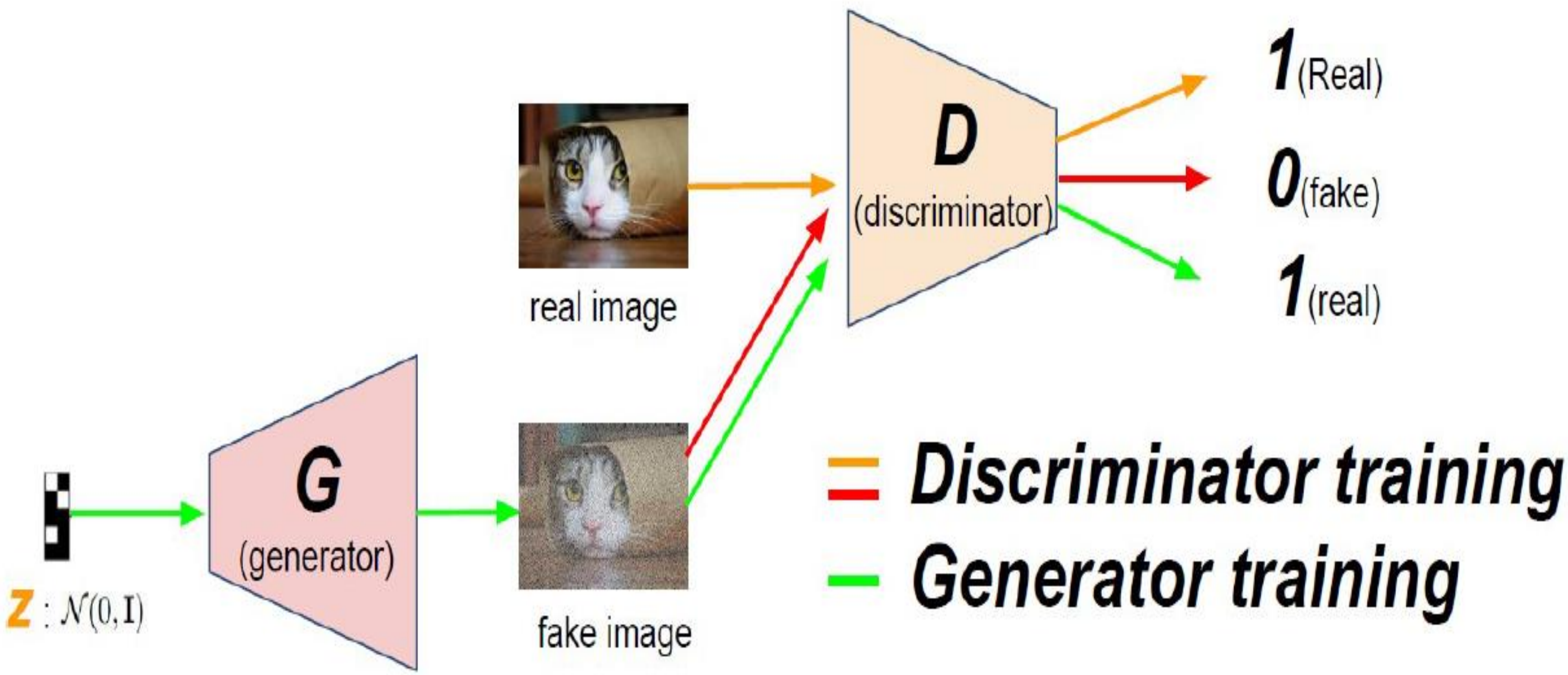
GAN을 활용한 이미지-이미지 변환

- 이미지-이미지 변환 (Pix2Pix)

pixel2pixel '16.11 (https://affinelayer.com/pixsrv/index.html)

GAN을 활용한 이미지-이미지 변환

- 복습: GAN의 학습



GAN을 활용한 이미지-이미지 변환

- 생성자 네트워크 구현

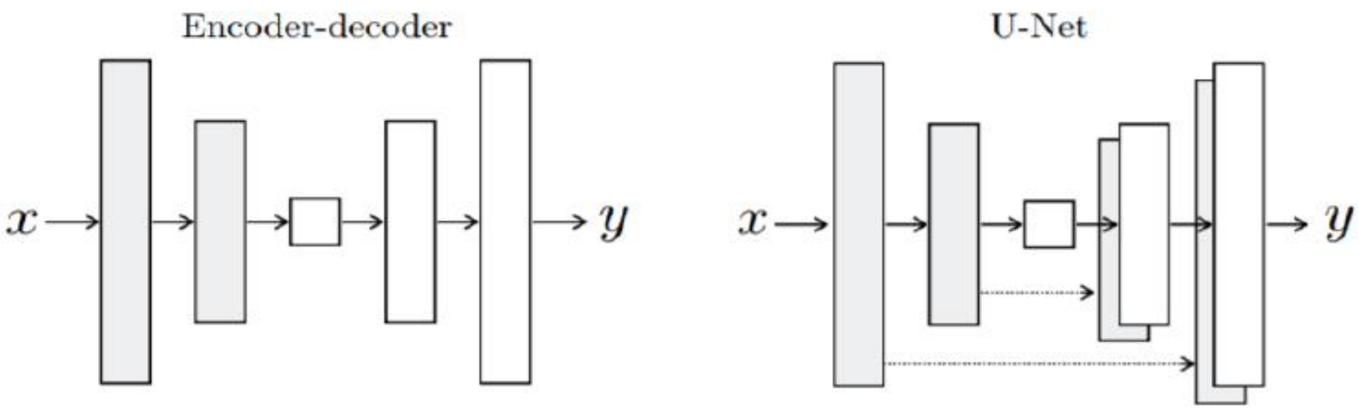


Figure 3: Two choices for the architecture of the generator. The "U-Net" [50] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

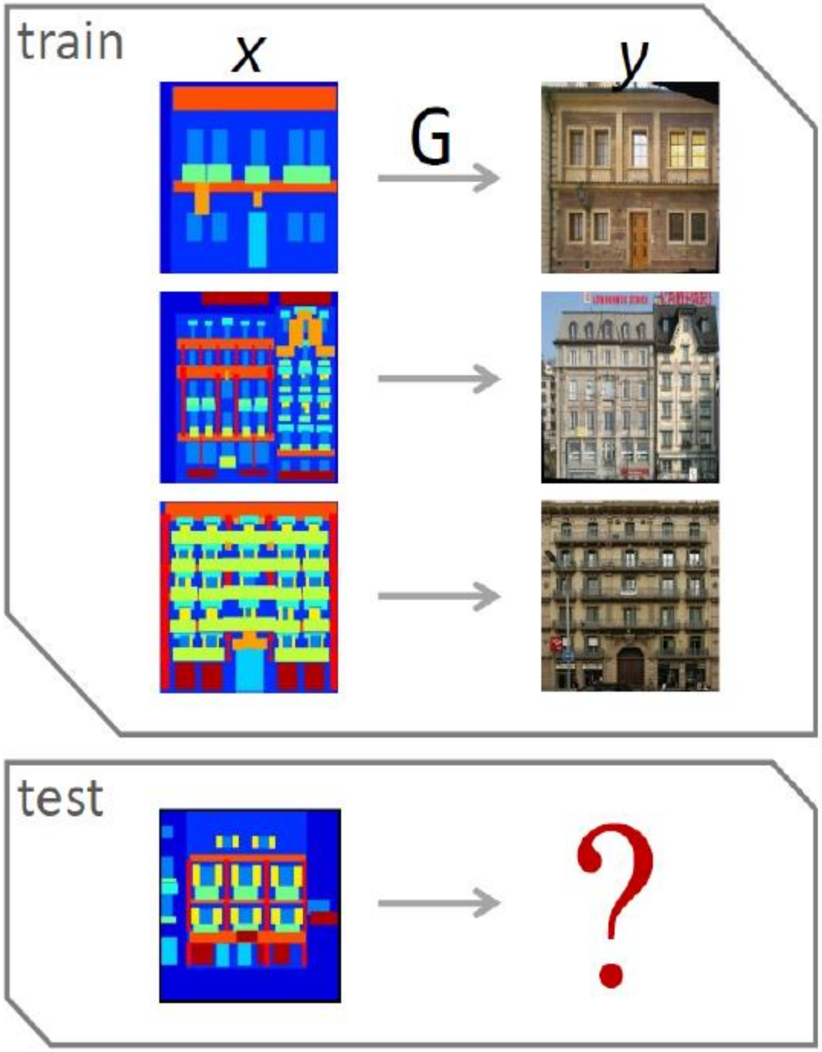
+ L1 loss function **Low-freq correctness**

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

+ PatchGAN **High-freq correctness**

GAN을 활용한 이미지-이미지 변환

- 생성자 네트워크 구현



- Supervised
- loss: Minimize the difference between output *G(x)* and ground truth *y*

Data from [Tylecek, 2013]

GAN을 활용한 이미지-이미지 변환

- 생성자 네트워크 구현: L1 손실

Loss: Minimize the difference between output *G(x) and the ground truth y*

$$\sum_{(x,y)} \|y - G(x)\|_1$$

Input Output Ground Truth

Input Output Ground Truth

GAN을 활용한 이미지-이미지 변환

- 생성자 손실함수: GAN 손실 + L1 손실

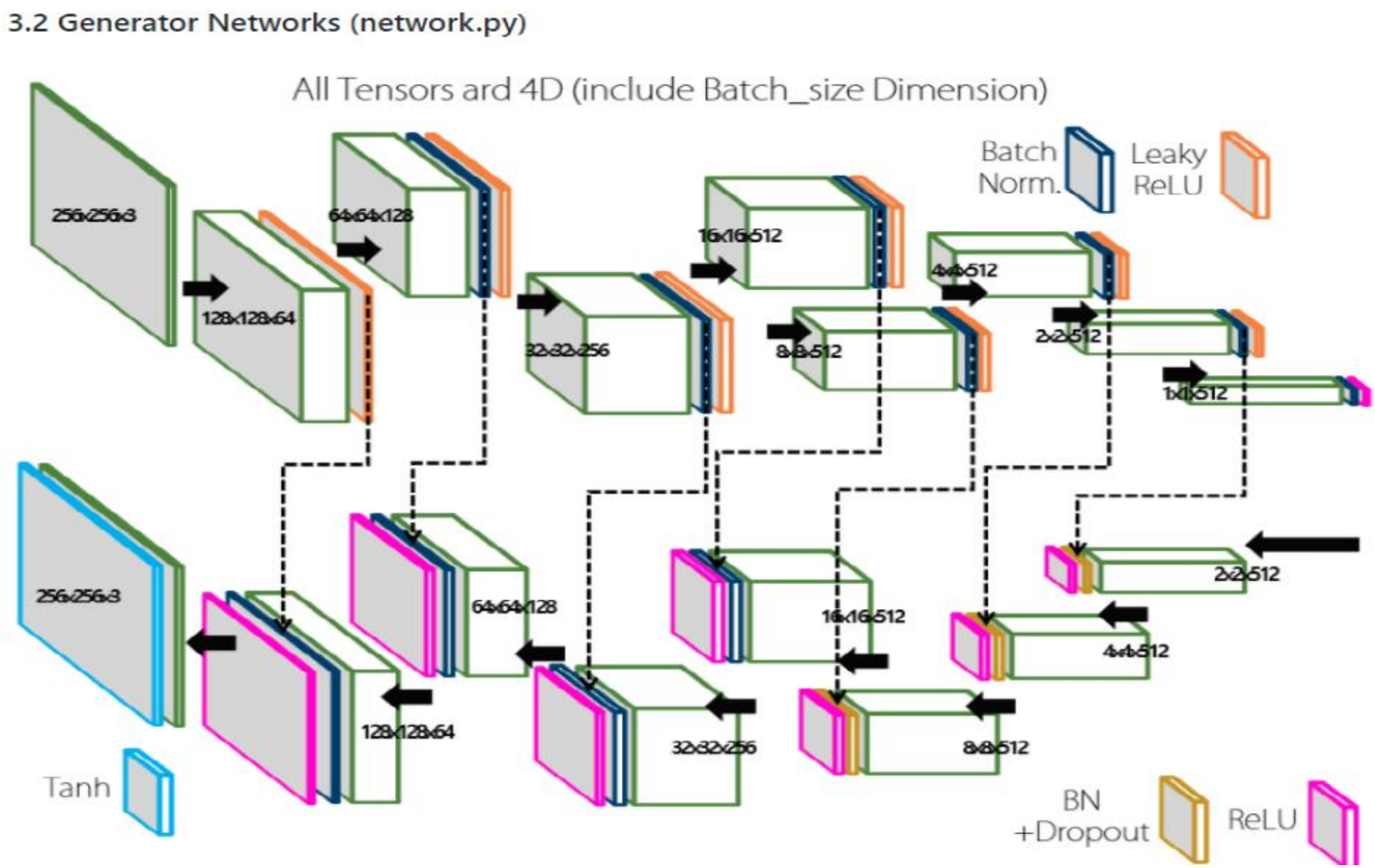
Loss: Minimize the difference between and output *G(x)* and ground truth *y*

$$\sum_{(x,y)} \|y - G(x)\|_1 + L_{GAN}\,(G(x), y)$$

Input　　Ground Truth　　L1 loss only　　L1+GAN loss

GAN을 활용한 이미지-이미지 변환

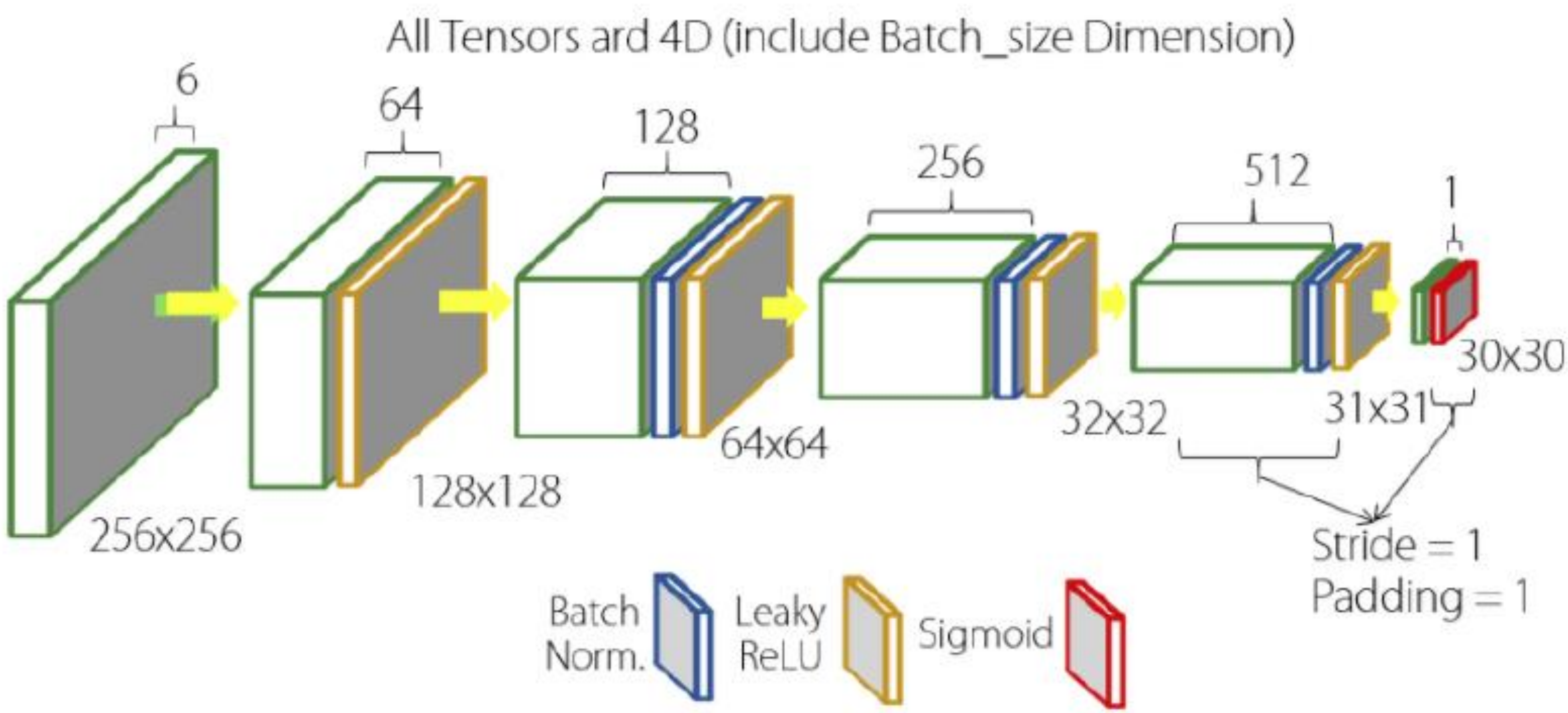
- Pix2Pix 생성자 네트워크 구조



GAN을 활용한 이미지-이미지 변환

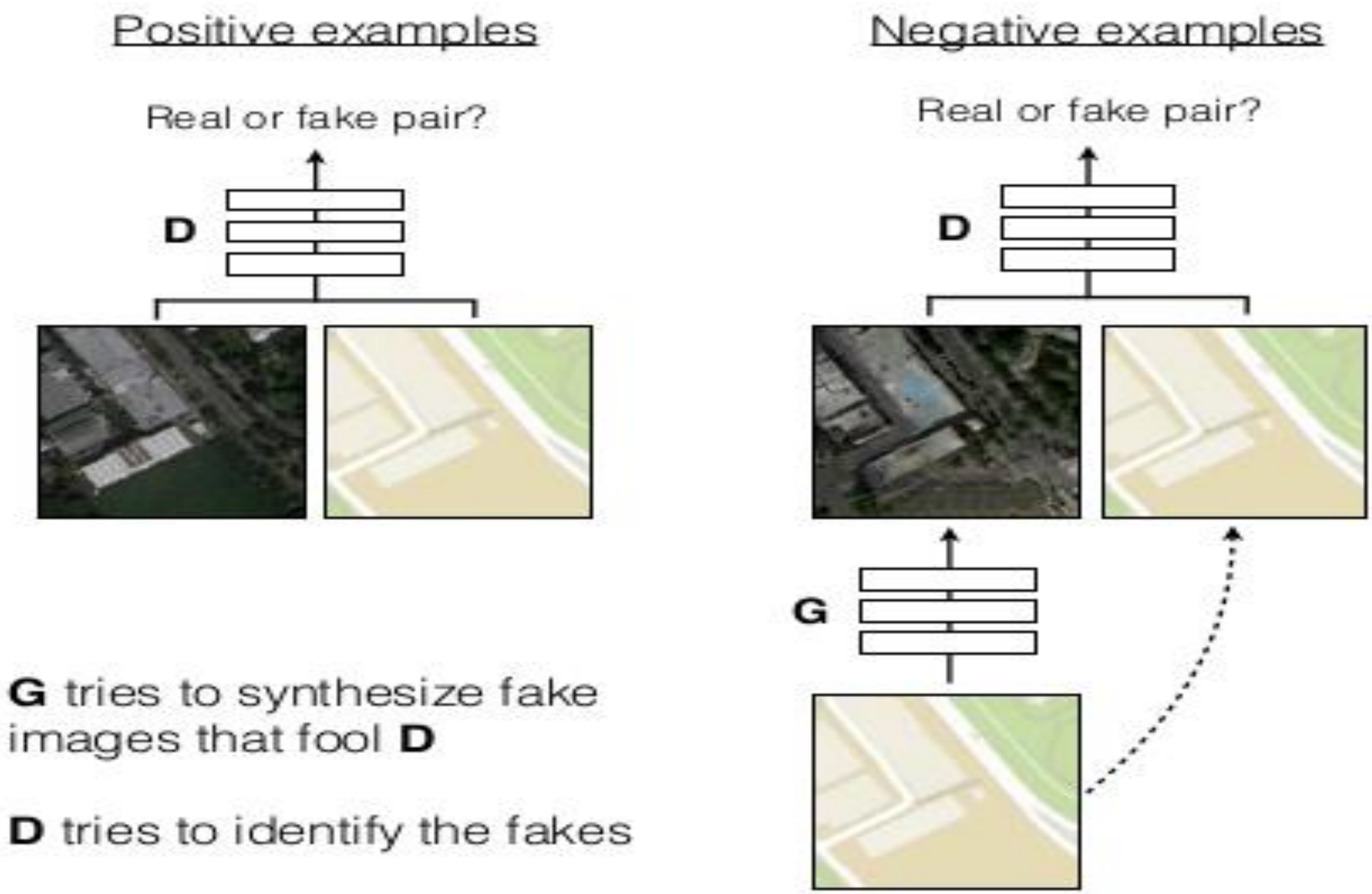
- Pix2Pix 판별자 네트워크 구조

3.3 Discriminator Networks (network.py)



GAN을 활용한 이미지-이미지 변환

- 이미지-이미지 변환 (Pix2Pix)



GAN을 활용한 이미지-이미지 변환

- 이미지 변환 사례

Figure 8: Example results on Google Maps at 512x512 resolution (model was trained on images at 256x256 resolution, and run convolutionally on the larger images at test time). Contrast adjusted for clarity.

Figure 11: Example results of our method on Cityscapes labels→photo, compared to ground truth.

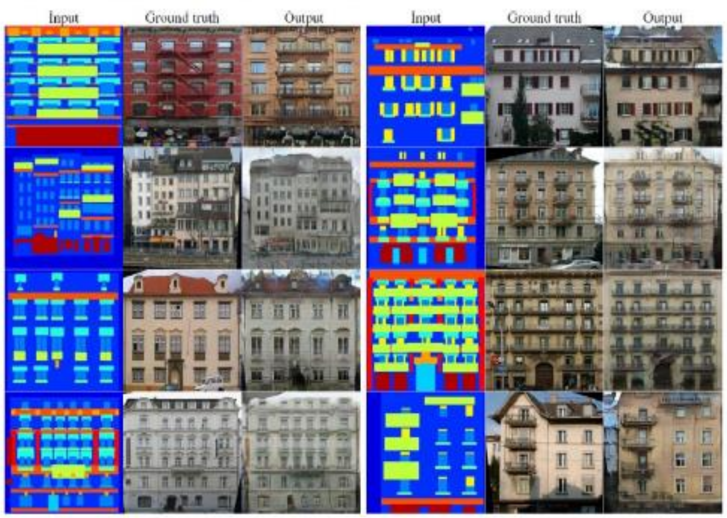


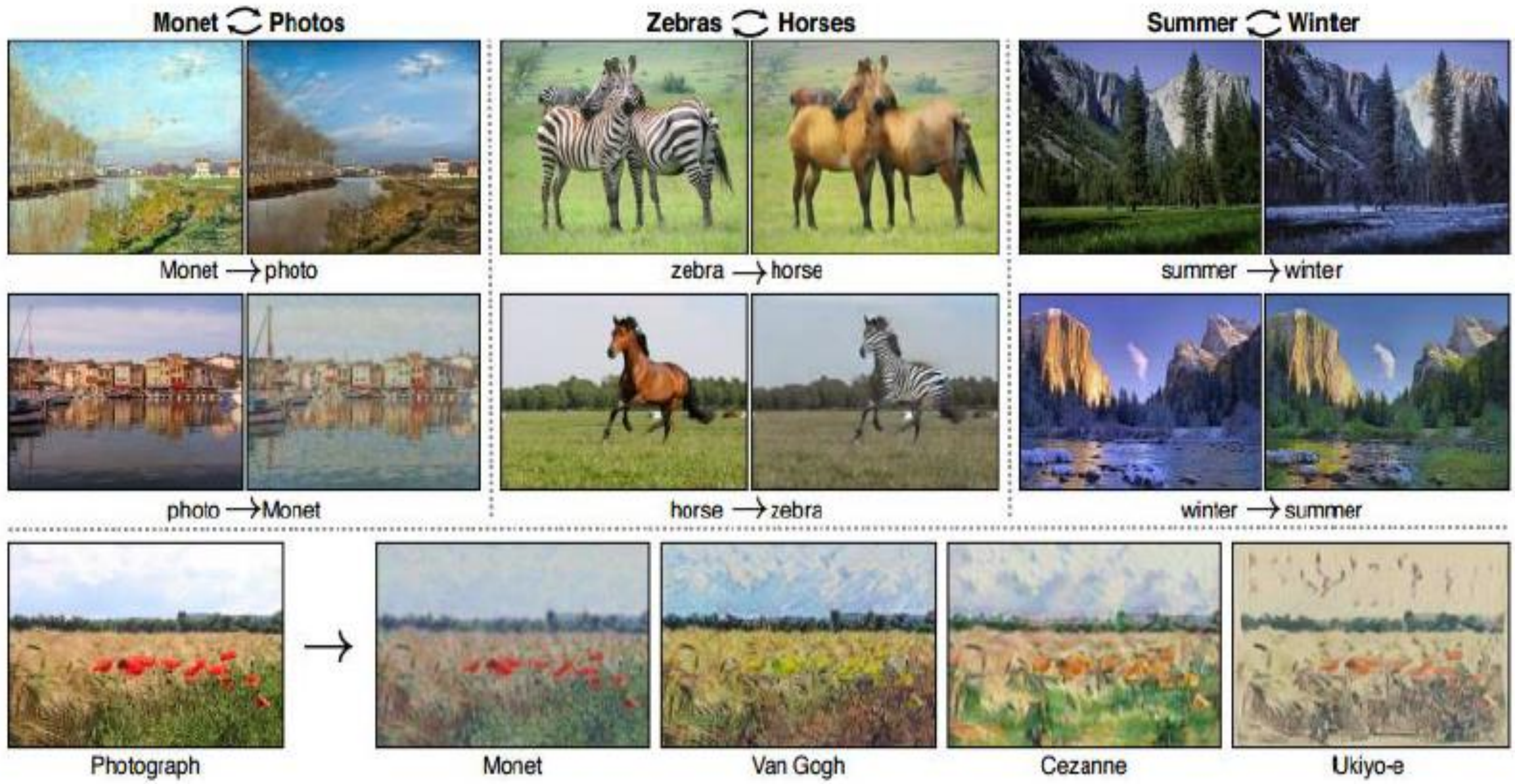
Figure 12: Example results of our method on facades labels→photo, compared to ground truth

Figure 13: Example results of our method on day→night, compared to ground truth.

2

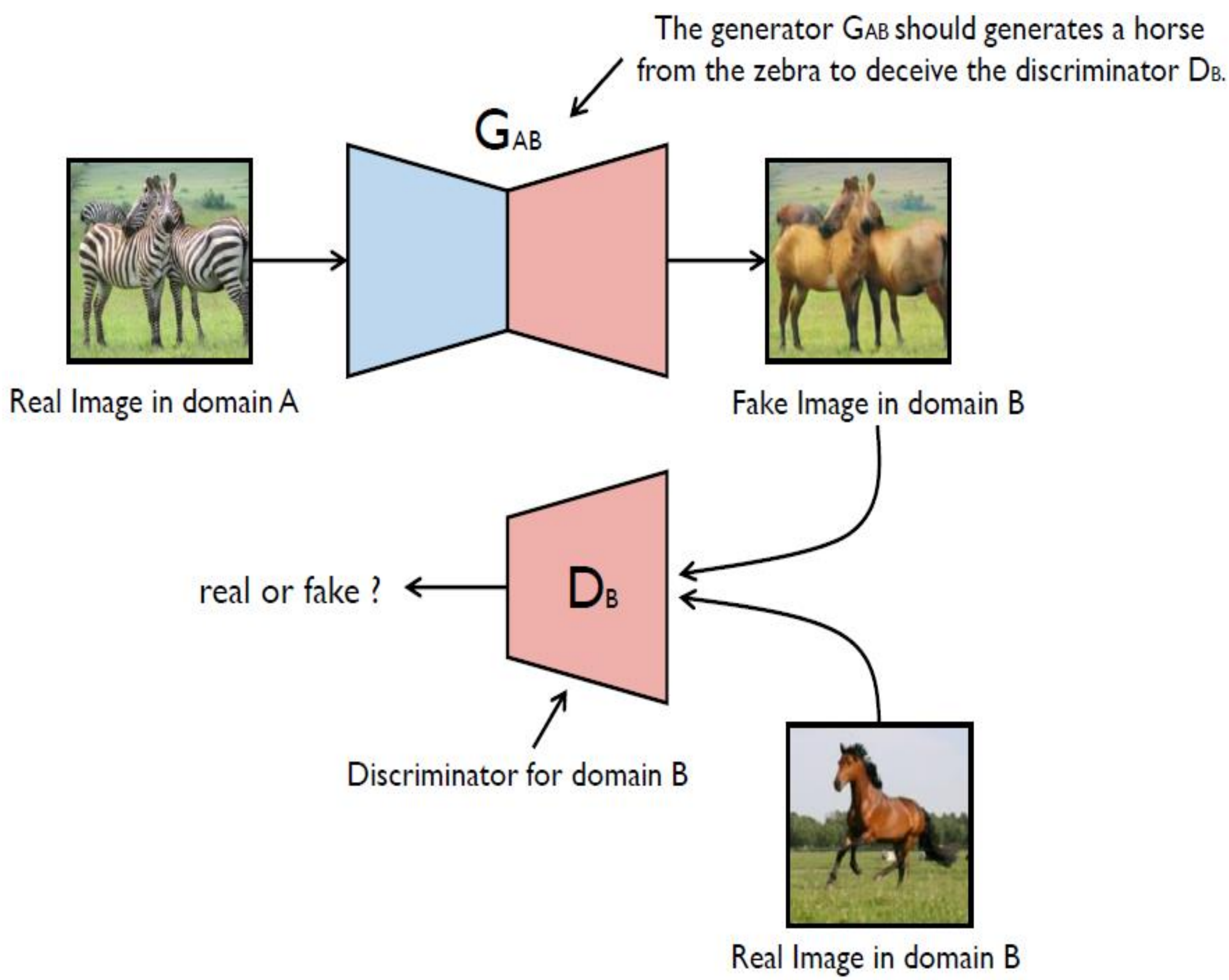
Cycle GAN을 활용한 이미지 변환

- CycleGAN: Image to Image Translation



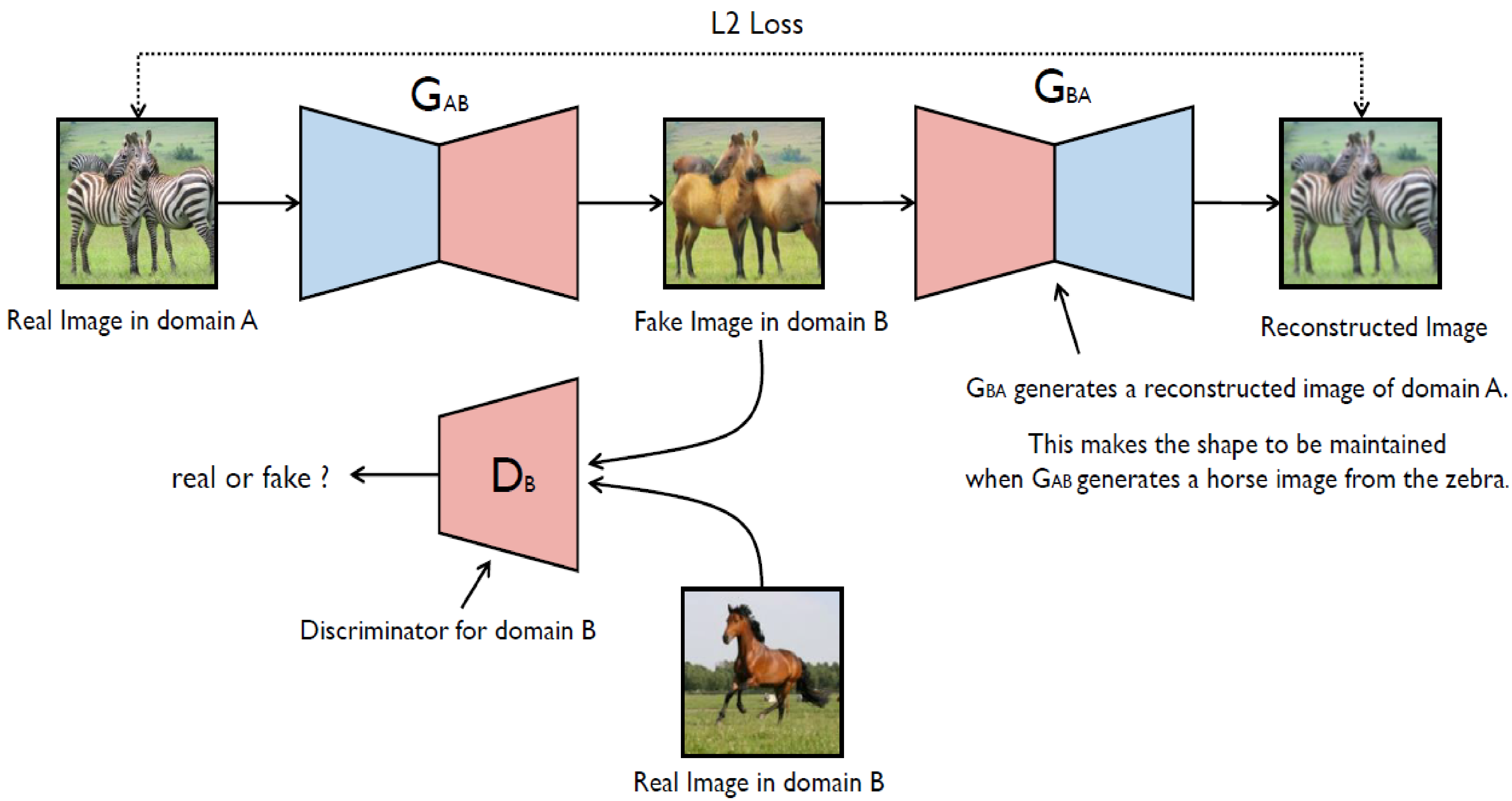
Cycle GAN을 활용한 이미지 변환

- CycleGAN



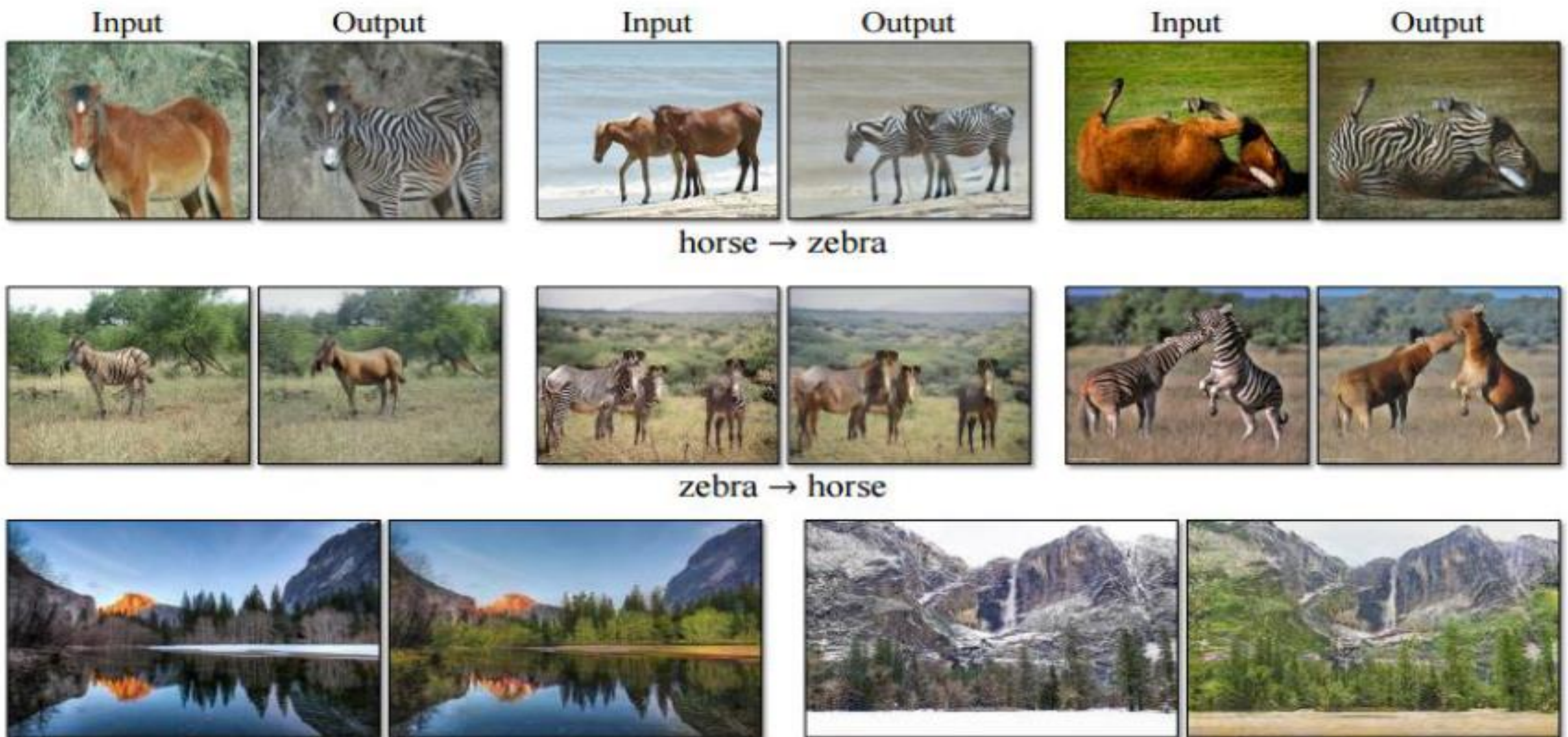
Cycle GAN을 활용한 이미지 변환

- CycleGAN



Cycle GAN을 활용한 이미지 변환

- CycleGAN



winter Yosemite → summer Yosemite

StackGAN을 활용한 텍스트-이미지 생성

- StackGAN: text to image