UNIT-II

Part – II

Major Architectures of Deep Networks

1

Generative Adversarial Networks (GANs)

Topics :

- Generative Adversarial Network
  - Architecture of GAN
  - Mathematical Notation
  - Loss Function GAN
  - Training Process of GAN
  - Types of GAN
  - Applications of GAN
  - Future generations of GAN
  - Differences b/w VAN and GAN

Generative Adversarial Networks (GANs)

- Generative Adversarial Networks (GANs) are a powerful class of neural networks that are used for unsupervised learning. It was developed and introduced by Ian J. Goodfellow in 2014.
- Generative modeling generates unstructured data such as new images or text or Videos.
- GAN is a class of algorithmic Deep learning framework having two neural networks that connect and can analyze, capture and copy the variations within a dataset.

Contd..

- To understand the **term GAN let's break it into separate three separate Words.** and each of them has its **separate meaning**, which is as follows:

1. Generative – To learn a generative model, which describes how data is generated in terms of a probabilistic model. In simple words, it **explains how data is generated visually.**

2. Adversarial – Adversarial in GANs means that **two networks** — the **Generator (G) and the Discriminator (D)** — are trained in opposition to each other. They are involved in a minimax game, where:

- The **Generator** tries to create fake data that **looks real.**
- The **Discriminator** tries to detect whether data is real or fake.

CONTD..

3. **Networks –** Use **deep neural networks** as artificial intelligence (AI) algorithms for **training purposes**.

GANs are a type of neural network architecture that can **generate new data based on the patterns** learned from a **given dataset**.

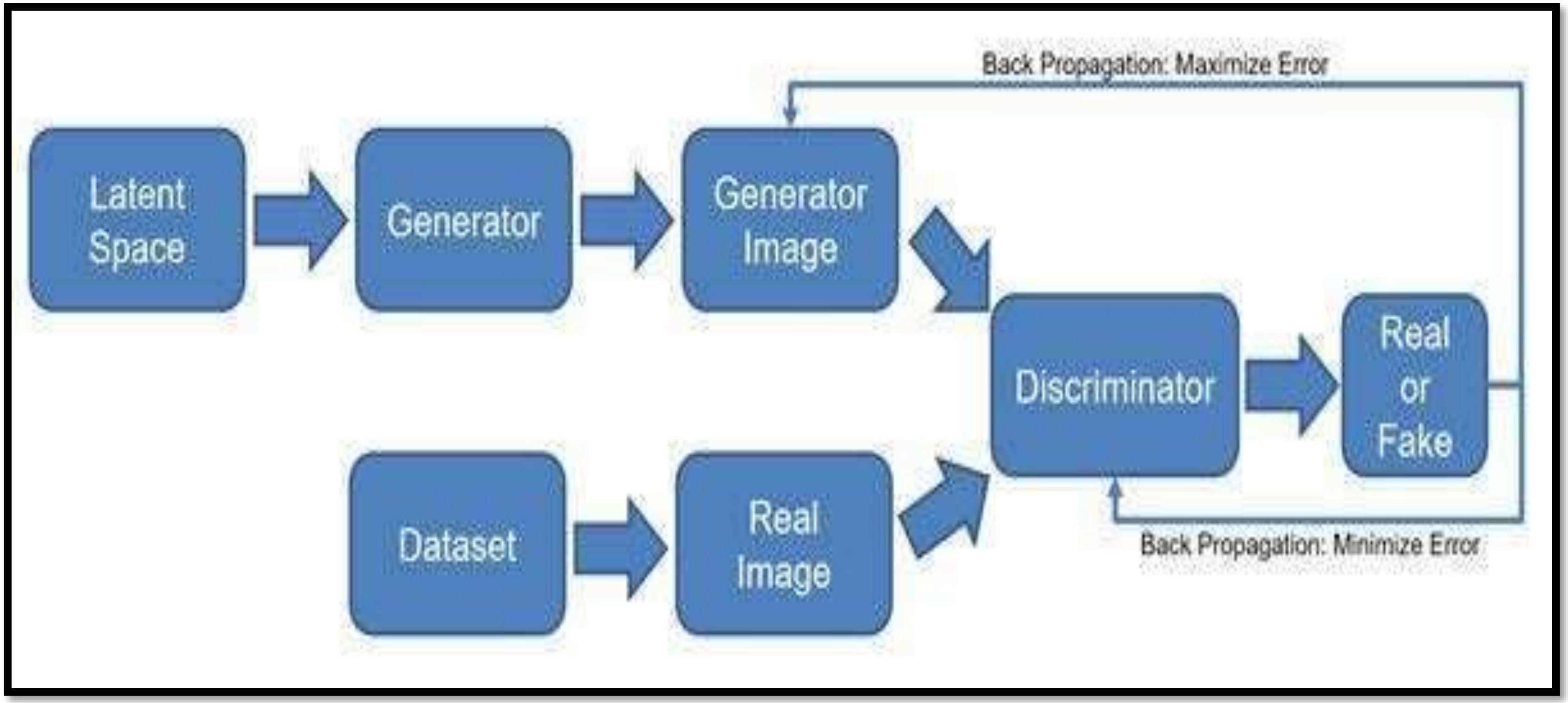
This means that GANs can create **entirely new, realistic images, videos, and even audio clips** that have **never existed before.**

ARCHITECTURE OF GAN

- **Generative Adversarial Networks (GANs)** are a groundbreaking innovation in the field of **Deep Learning**, particularly within the **domain of unsupervised learning**.
- GANs are made up of two neural networks,
  - **1. Generator and**
  - **2. Discriminator**
- These **2 models** that **automatically discover** and **learn the patterns in input data**.

Contd..

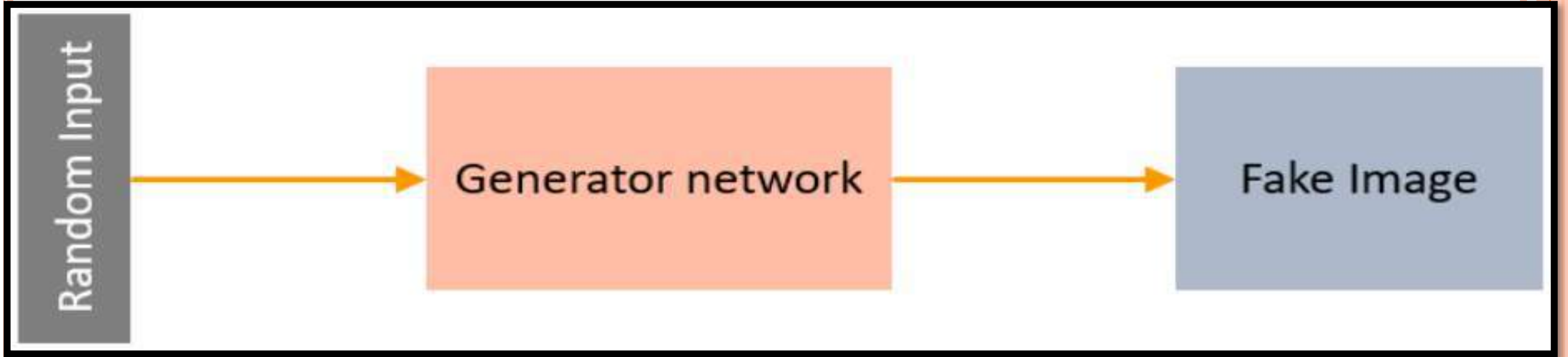
- They comprise two networks:
- The generator, which produces synthetic data. i.e, information that is artificially generated rather than produced by real-world events and
- The discriminator, which differentiates between real and generated data. This unique structure enables GANs to generate highly realistic and diverse outputs, from images to text.



The GAN Network Process

The Generator: The generator network is responsible for generating new data that is similar to the training data. The generator network takes random noise as input and produces a generated output. The goal is to train the generator to produce outputs that are as **close to the real data as possible.**

- A Generator in GANs is a neural network that creates fake data to be trained on the discriminator.



Contd..

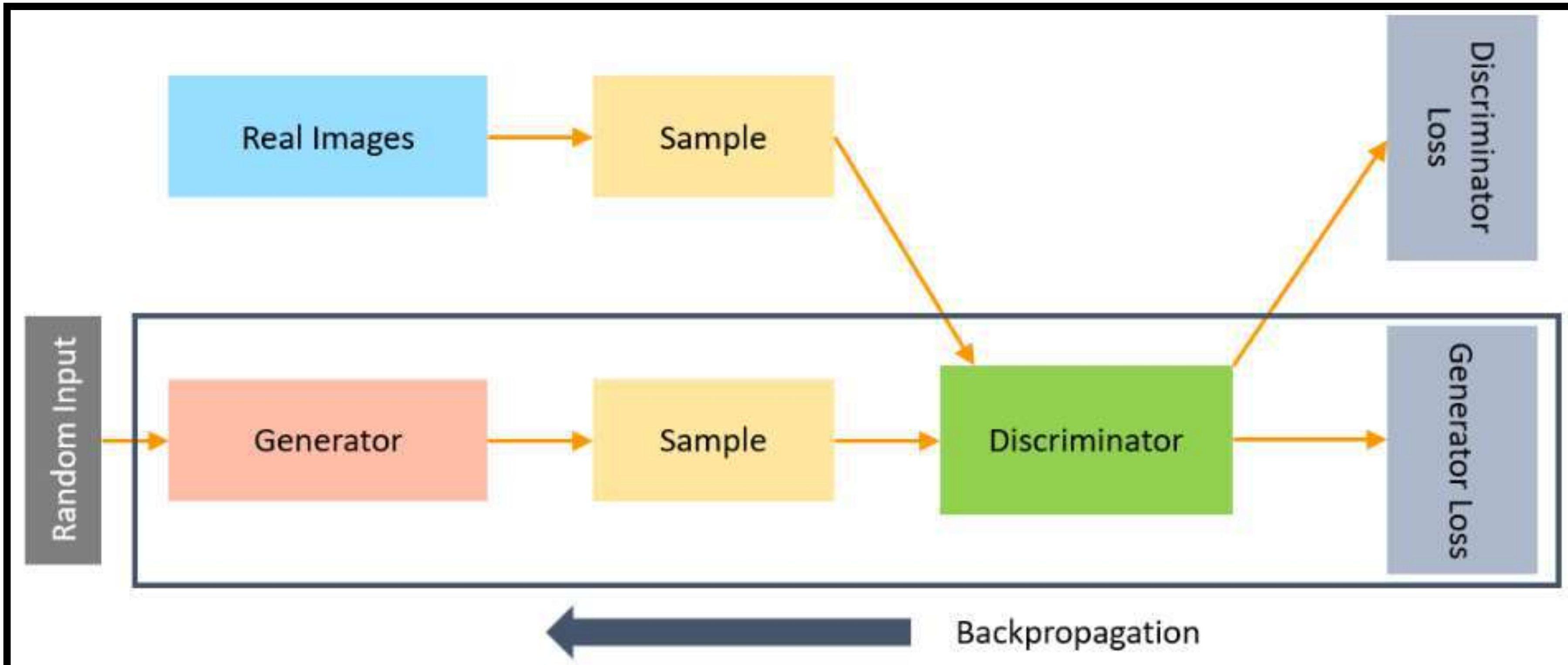
- The main aim of the Generator is to make the discriminator classify its output as real. The part of the GAN that trains the Generator includes:
  - Provide fake input or noise and get random noise to produce output based on the noise sample.
  - Predict generator output either **real or fake** using discriminator.
  - Calculate discriminator loss and perform back propagation.
  - Calculate gradients to update the weights of the generator.

CONTD..

- Back propagation of Generator: The generator modifies some data attributes by **adding noise (or random changes)** to certain attributes
- The **generator passes** the **modified data** to the **discriminator**
- The **discriminator calculates** the **probability** that the **generated output** bel**ongs to the original dataset**
- The discriminator gives some guidance to the generator to **reduce the noise vector randomization in the next cycle**
- The **generator attempts to maximize the probability of mistake** by the **discriminator**, but the **discriminator attemp**ts **to minimize the probability of error.**

CONTD..

- In training iterations, both the generator and discriminator iterating continuously until they reach an equilibrium state. In the equilibrium state, the discriminator can no longer recognize synthesized data. At this point, the training process is over.



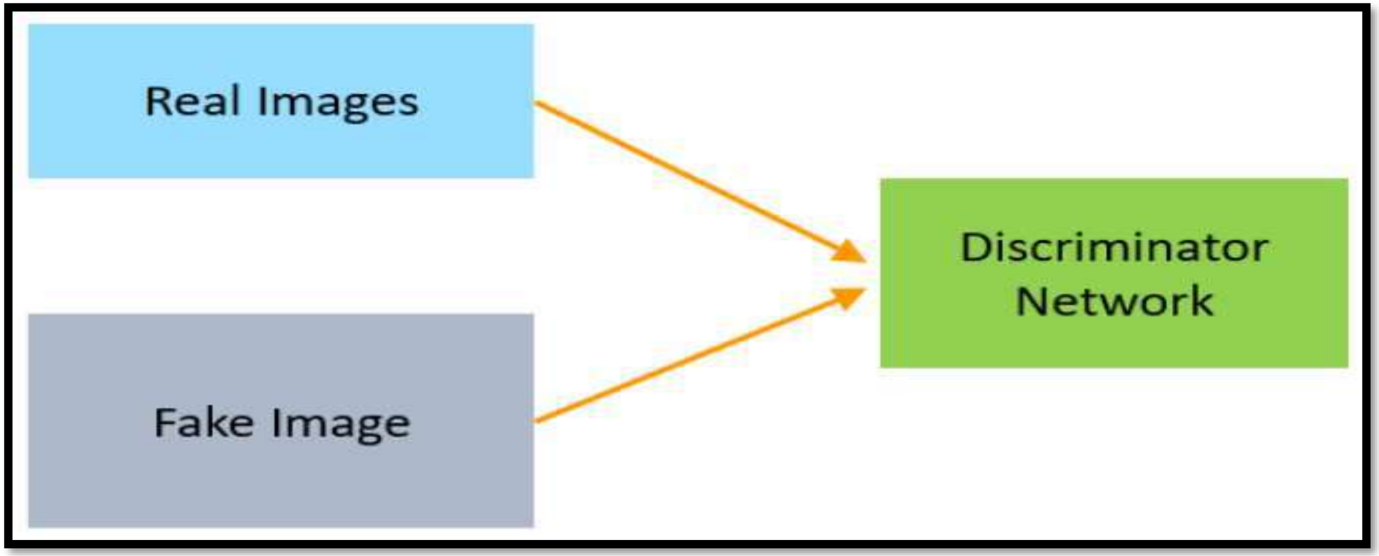
The Discriminator: In a GAN, the discriminator acts as a **binary classifier** whose main task is to differentiate between **real data** and **data generated by the GAN's generator**.

The **Discriminator is a neural network** that **identifies real data from the fake data created by the Generator. The discriminator's training data comes from different two sources:**

- The **real data instances**, such as **real pictures of birds, humans, currency notes, etc.**, are **used by the Discriminator** as **positive samples during training.**

Contd..

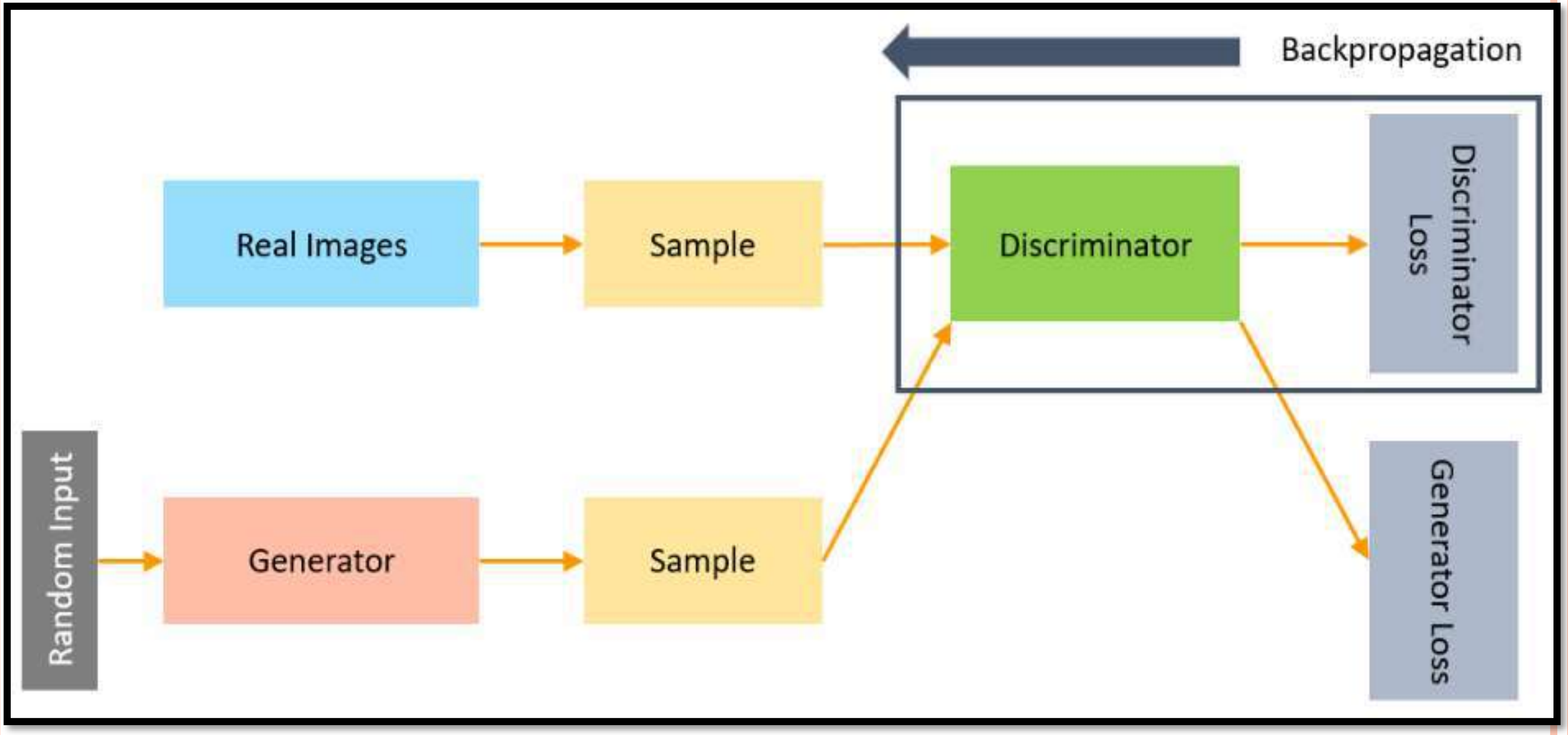
- The fake data instances created by the Generator are used as negative examples during the training process.



While training the discriminator, it connects to two loss functions. During discriminator training, the discriminator ignores the generator loss and just uses the discriminator loss.

Contd..

- Next, The discriminator updates its weights through backpropagation from the discriminator loss through the discriminator network.



Mathematical notation

- We are basically training the Discriminator to maximize the probability of assigning correct labels to **both real and generated data.**

  We are also training the Generator to minimize the probability to get caught by the Discriminator, which is **equivalent to minimizing** log(1-D(G(z))).

- here the Discriminator is trying to minimize its reward V(D, G) and the Generator is trying to minimize the Discriminator's reward or in other words, maximize its loss.

MATHEMATICAL NOTATION

$$\min_{G} \max_{D} V(D, G)$$

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

where, G = Generator, D = Discriminator

Pdata(x) = distribution of real data

P(z) = distribution of generator

x = sample from Pdata(x)

z = sample from P(z)

D(x) = Discriminator network

G(z) = Generator network

x : Real data

z : Latent vector

$G(z)$: Fake data

$D(x)$: Discriminator's evaluation of real data

$D(G(z))$: Discriminator's evaluation of fake data

Loss Functions in GAN

- The training process for GANs involves minimizing a loss function that **quantifies** the difference between the **generated and real data.**
- There are two types of Loss Functions:
  - 1. Generator Loss
  - 2. Discriminator Loss

1. Generator Loss: The **objective of the generator in a GAN** is to produce synthetic samples that are realistic **enough to fool the discriminator**. The generator achieves this by minimizing its loss function **J_G.**

Contd..

- The loss is minimized when the log probability is maximized, i.e., when the discriminator is highly likely to classify the generated samples as real. The following equation is given below:

$$J_G = -\frac{1}{m}\Sigma_{i=1}^{m} logD(G(z_i))$$

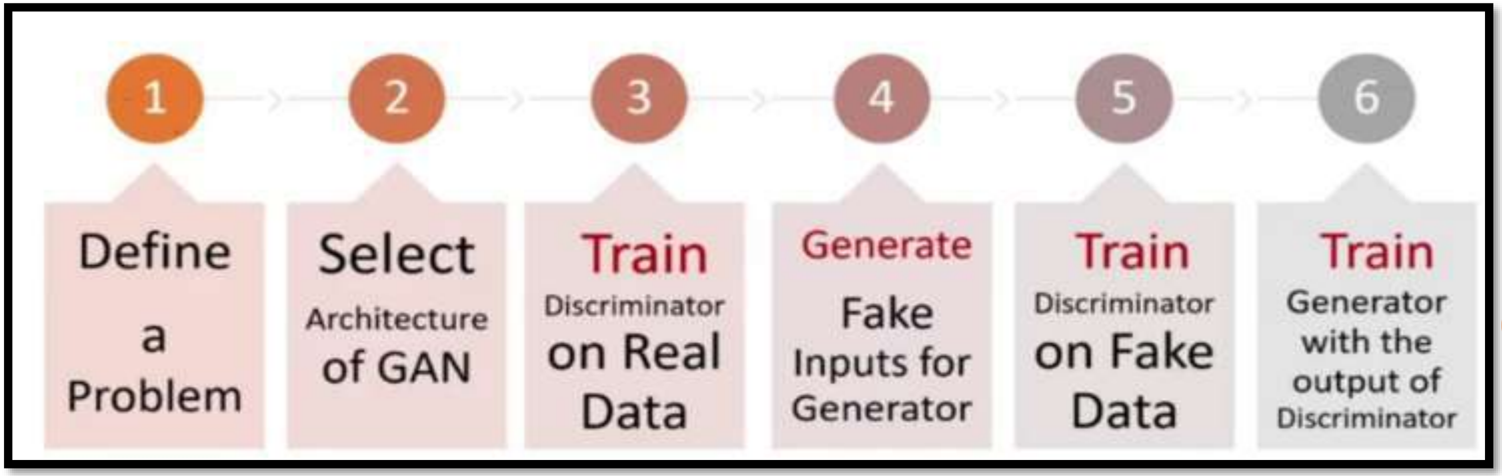
The generator aims to minimize this loss, encouraging the production of samples that the **discriminator classifies as real** (logD(G(zi)), close to 1.

2. Discriminator Loss: The discriminator **reduces the negative log likelihood** of correctly classifying both produced and real samples.

$$J_D = -\frac{1}{m}\Sigma_{i=1}^{m} log\, D(x_i) - \frac{1}{m}\Sigma_{i=1}^{m} log(1 - D(G(z_i)))$$

Training process of GAN

- **Step 1:** Define a Problem.
- **Step 2:** Select Architecture of GAN.
- **Step 3:** Train Discriminator on Real Dataset.
- **Step 4:** Train Generator.
- **Step 5:** Train Discriminator on Fake Data.
- **Step 6:** Train Generator with the output of Discriminator.



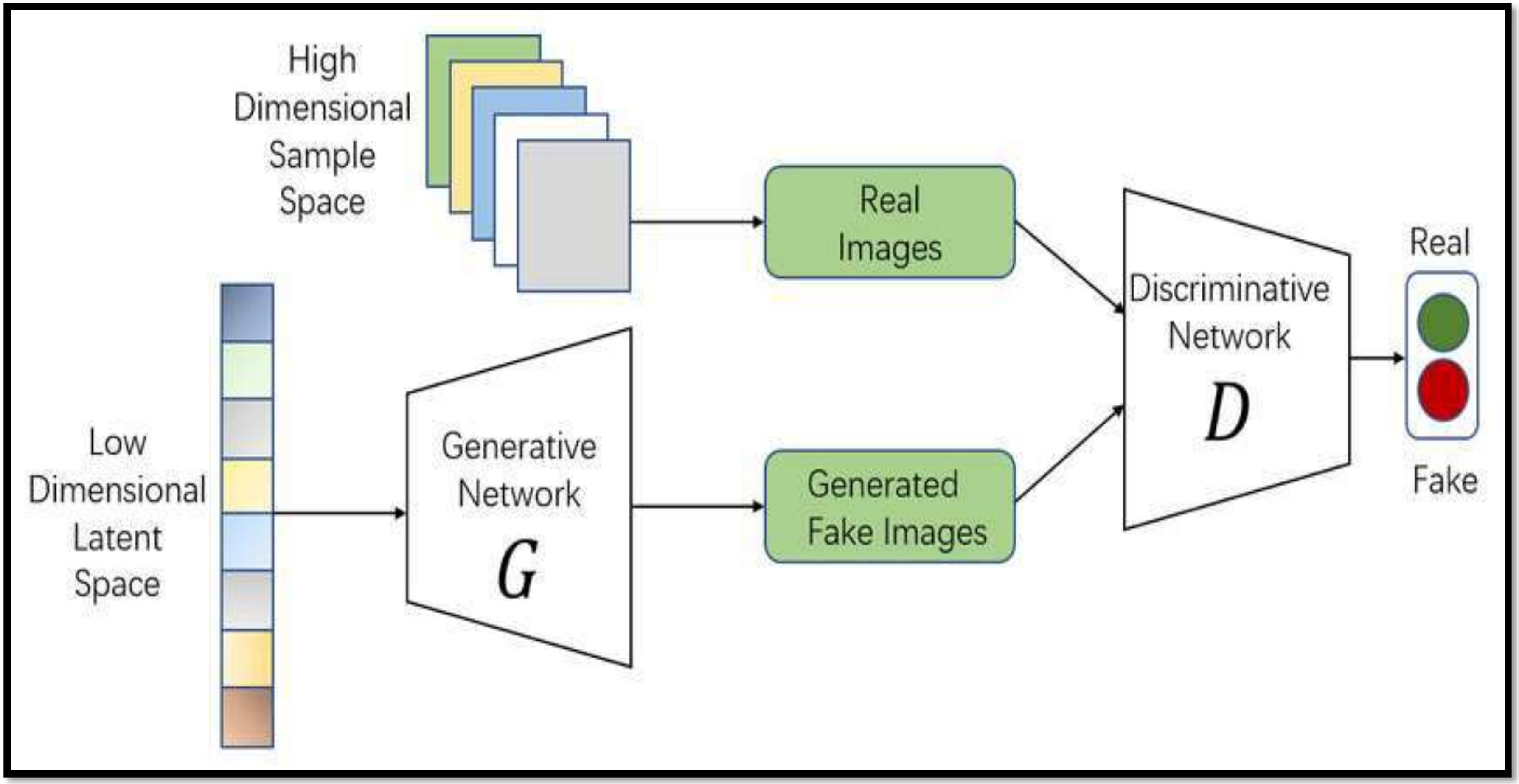
Types of GAN

Vanilla GAN
01
02
Conditional GAN (CGAN)
Deep Convolutional GAN (DCGAN)
03
04
CycleGAN
Generative Adversarial Text to Image Synthesis
05
06
StyleGAN
Super Resolution GAN (SRGAN)
07

CONTD..

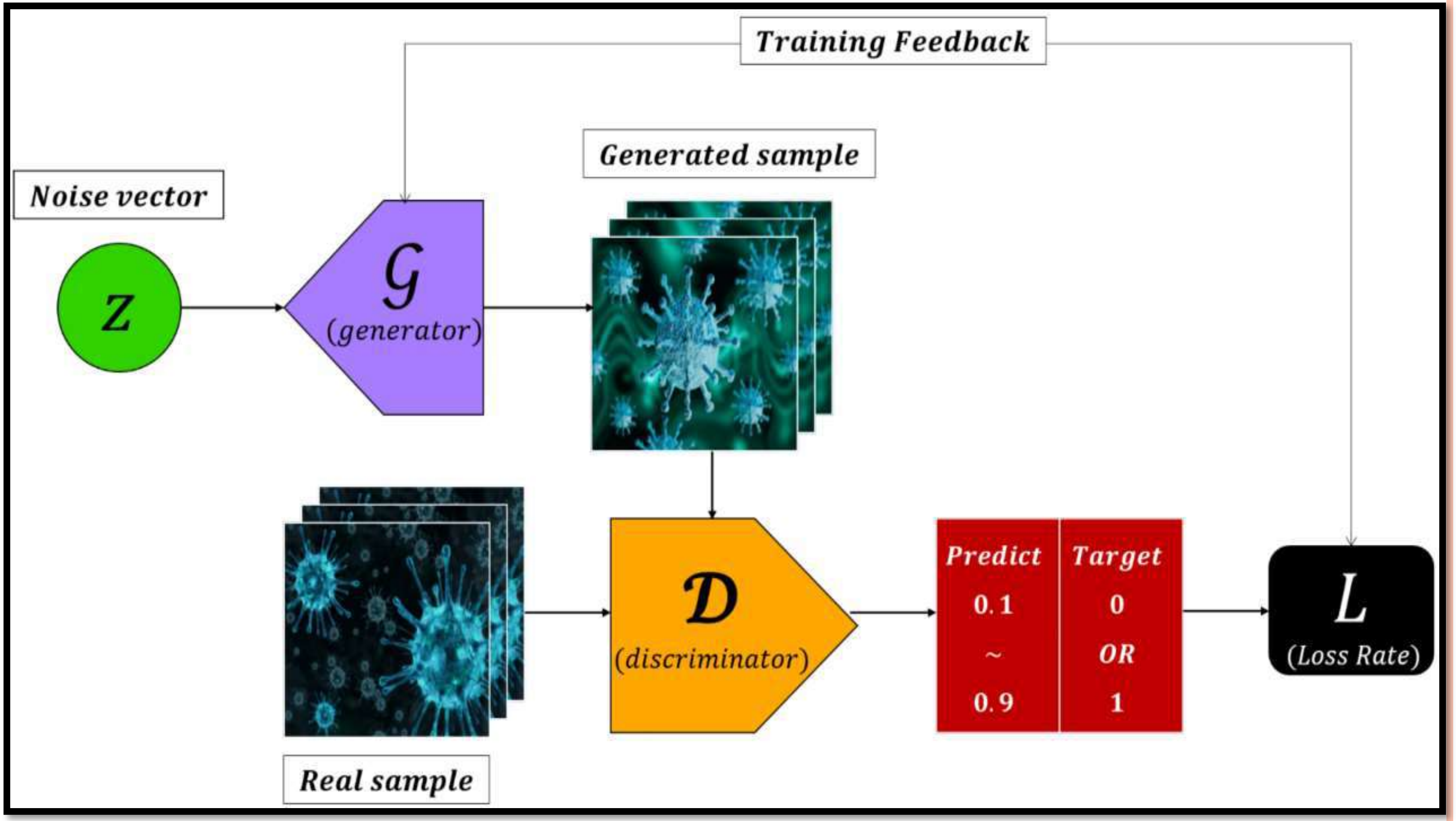
- **1. Vanilla GAN:** This is the most **basic form of GAN**, where **both the generator and discriminator** are modeled using **multi-layer perceptrons.**
- The **generator focuses on capturing data distribution**, while the **discriminator evaluates the probability that a given sample** is **real or synthetic.**
- This Vanilla GAN always **tries to optimize** the **mathematical equation** using **stochastic gradient descent.**

Contd..



ARCHITECTURE OF VANILLA GAN

Contd..

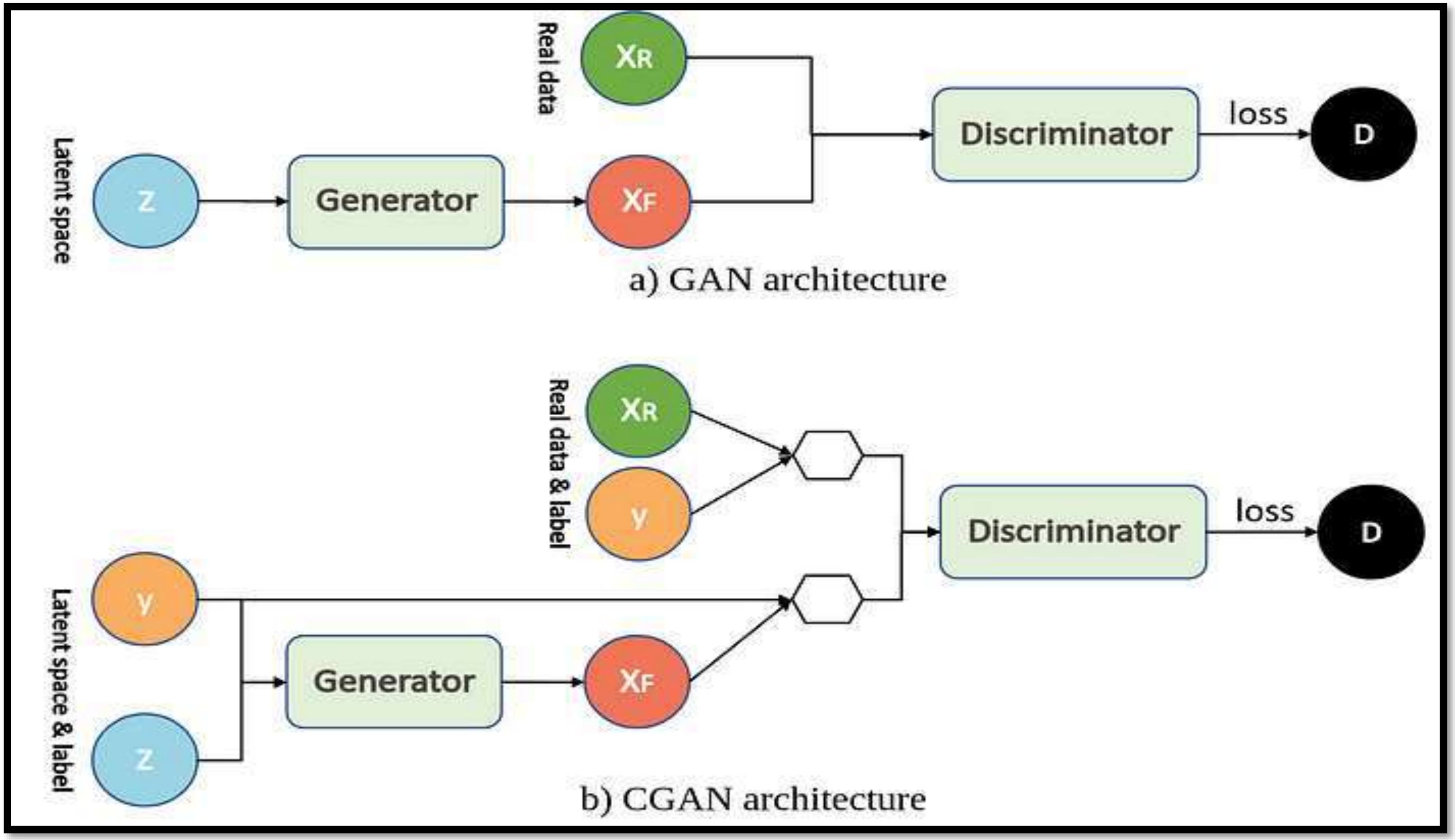


EXAMPLE OF VANILLA GAN

CONTD..

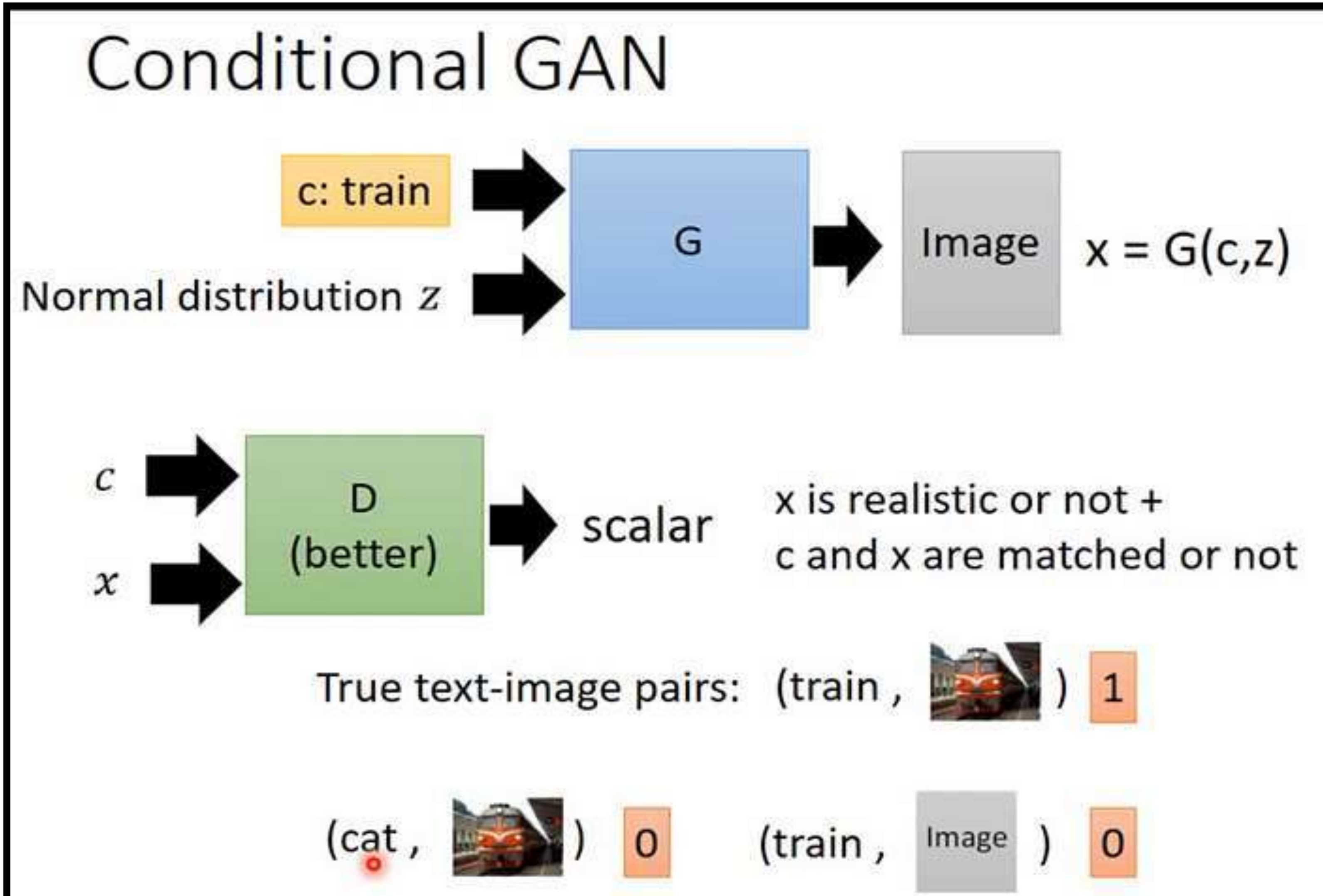
- **2. Conditional GAN:** CGAN can be described as a deep learning method in which **some conditional parameters** are **put into place.**
- Here, **both networks receive additional information** such as **class labels,** making the model conditional. This allows the **generation of data** that is more **specific to the given condition**.
- In CGAN, an **additional parameter 'y' is added** to **the Generator** for **generating the corresponding data.**
- **Labels** are also put into the **input to the Discriminator** in order for the **Discriminator to help distinguish the real data from the fake generated data.**

Contd..



Architecture of Conditional GAN (CGAN)

CONTD..



Example of Conditional GAN

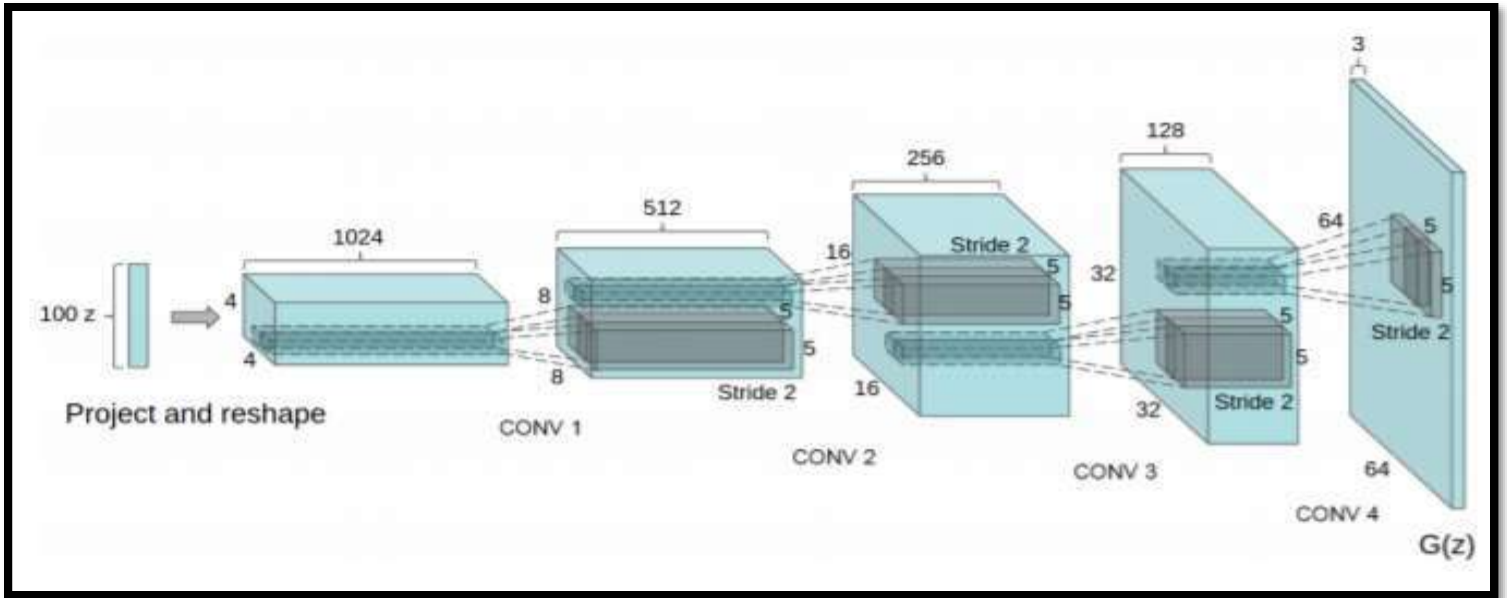
(CGAN)

Contd..

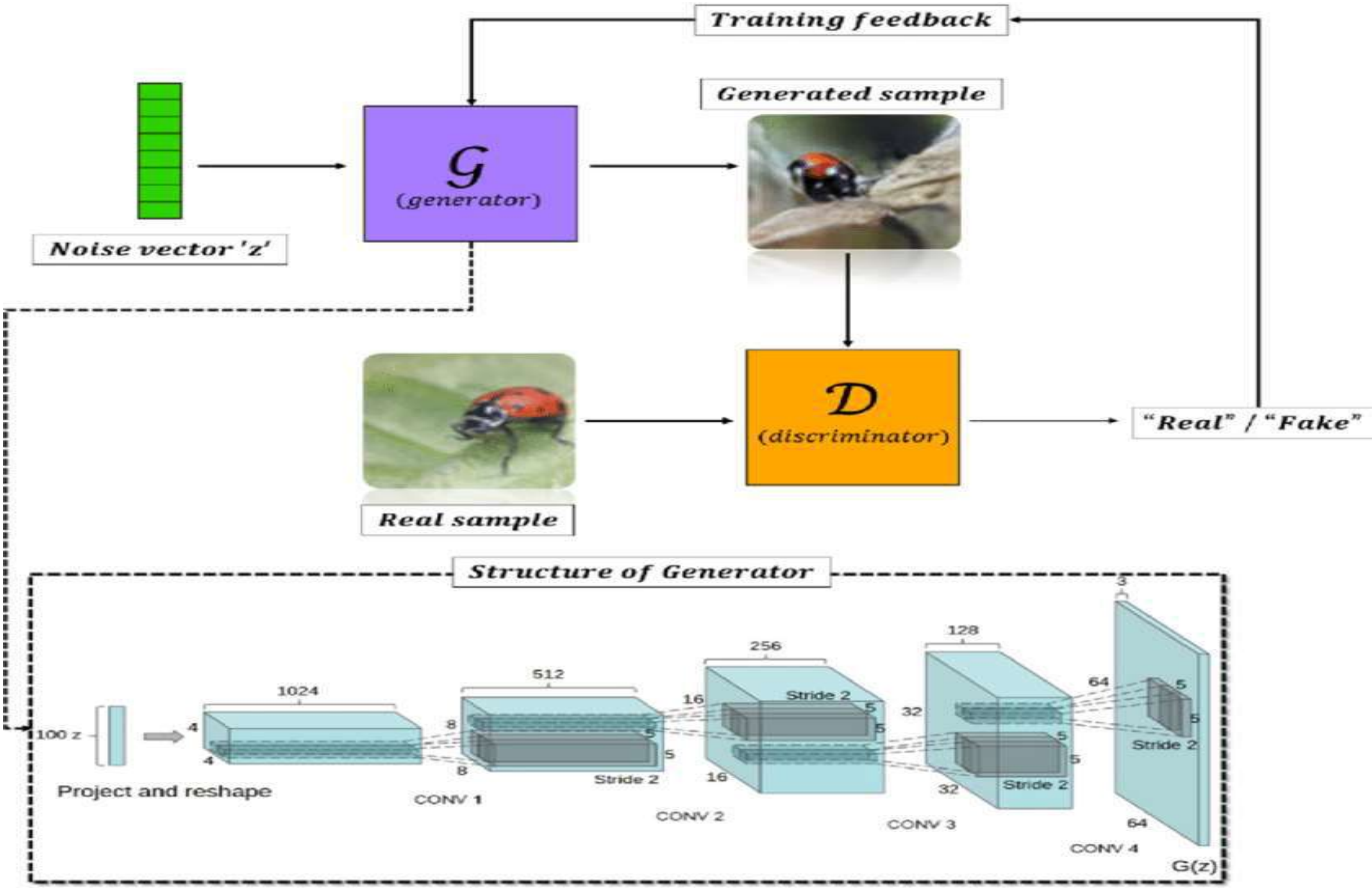
- 3. Deep Convolutional GAN (DCGAN): DCGANs employ convolutional neural networks, making them more effective for tasks that involve image data. They are known for generating high-quality, high-resolution images.
- Deep Convolutional GAN (DCGAN) was proposed by a researcher from MIT and Facebook AI research. It is widely used in many convolution-based generation-based techniques.
- It is composed of ConvNets in place of multi-layer perceptrons.
- DCGAN uses convolutional and convolutional-transpose layers in the generator and discriminator, respectively. It was proposed by Radford.

CONTD..

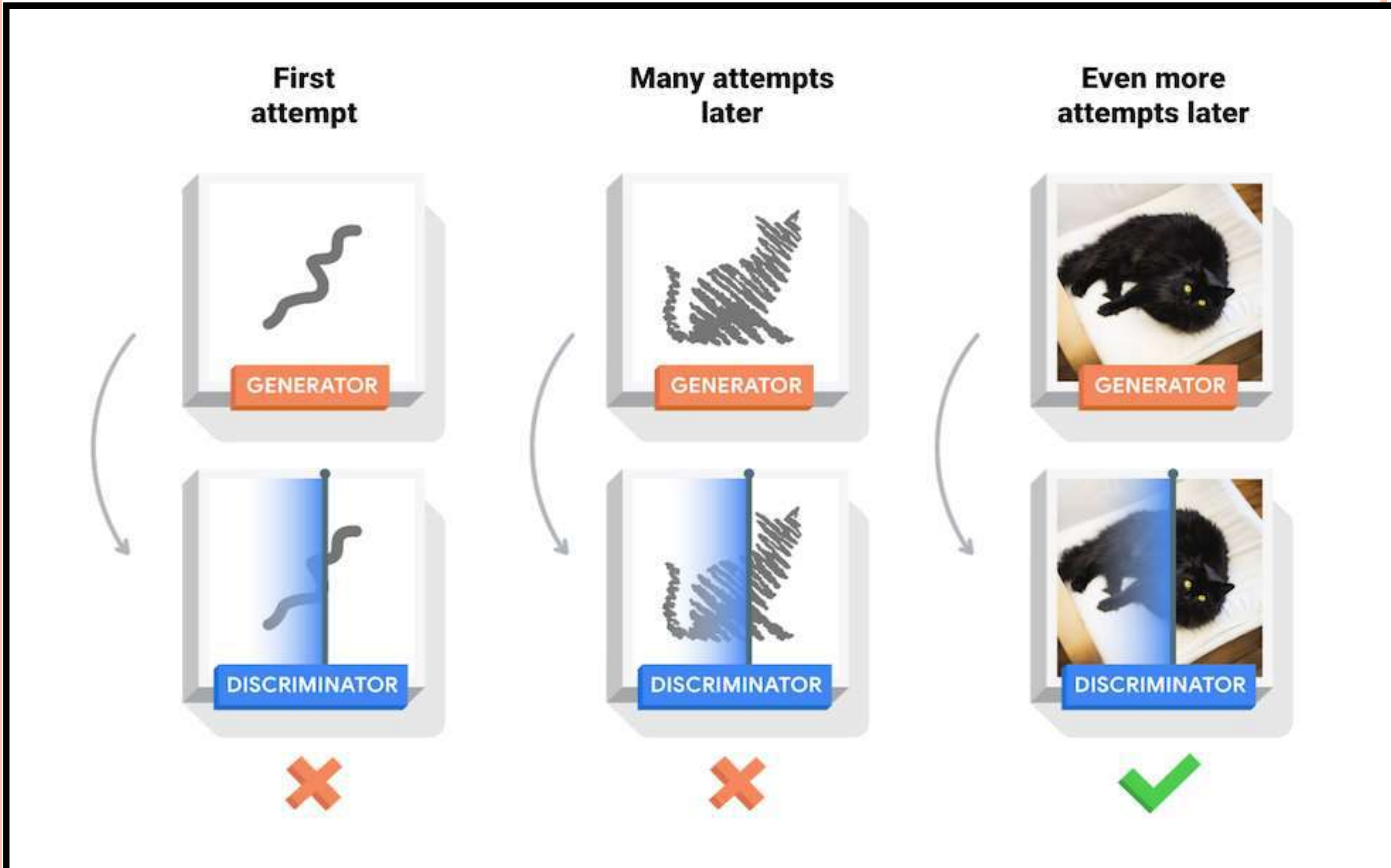
- Here the discriminator consists of strided convolution layers, and Relu as activation function.. The generator consists of convolutional-transpose layers, and ReLU activations. The output will be a 3x64x64 RGB image.



Architecture of Deep Convolution GAN (DCGAN)



Example of DCGAN (DCGAN)



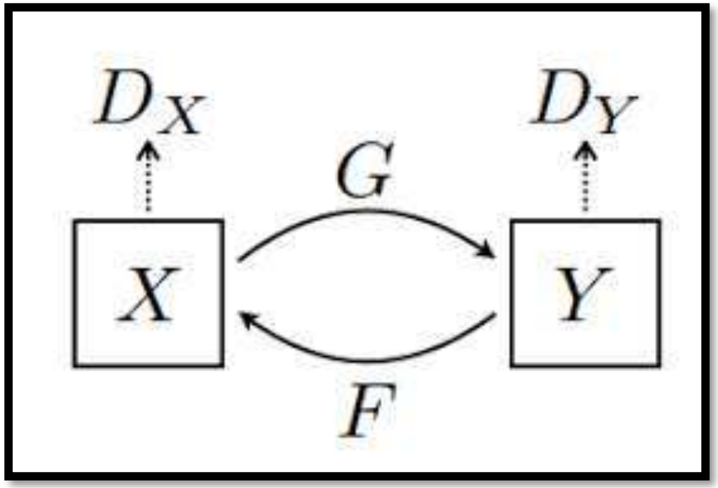
Example of DCGAN (DCGAN)

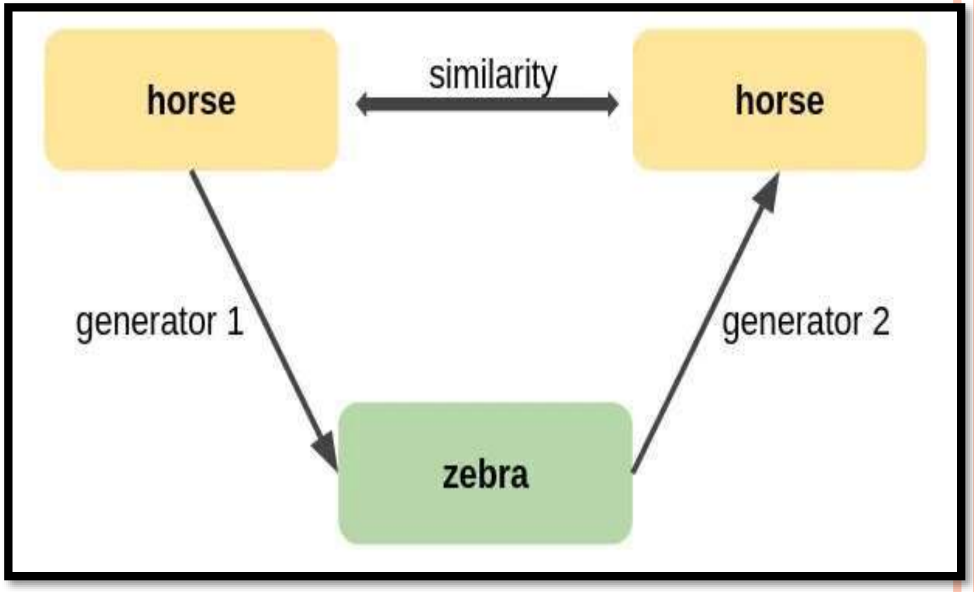
Contd..

- **4. Cycle GAN:** **Cycle GAN** is used to transfer characteristic of one image to another or can map the distribution of images to another.
- In **CycleGAN** we treat the problem as an image reconstruction problem. We first take an image input (x) and using the generator G to convert into the reconstructed image.
- Then we reverse this process from **reconstructed image to original image using a** generator F.
- Then we calculate the mean squared error loss between real and reconstructed image.

CONTD..

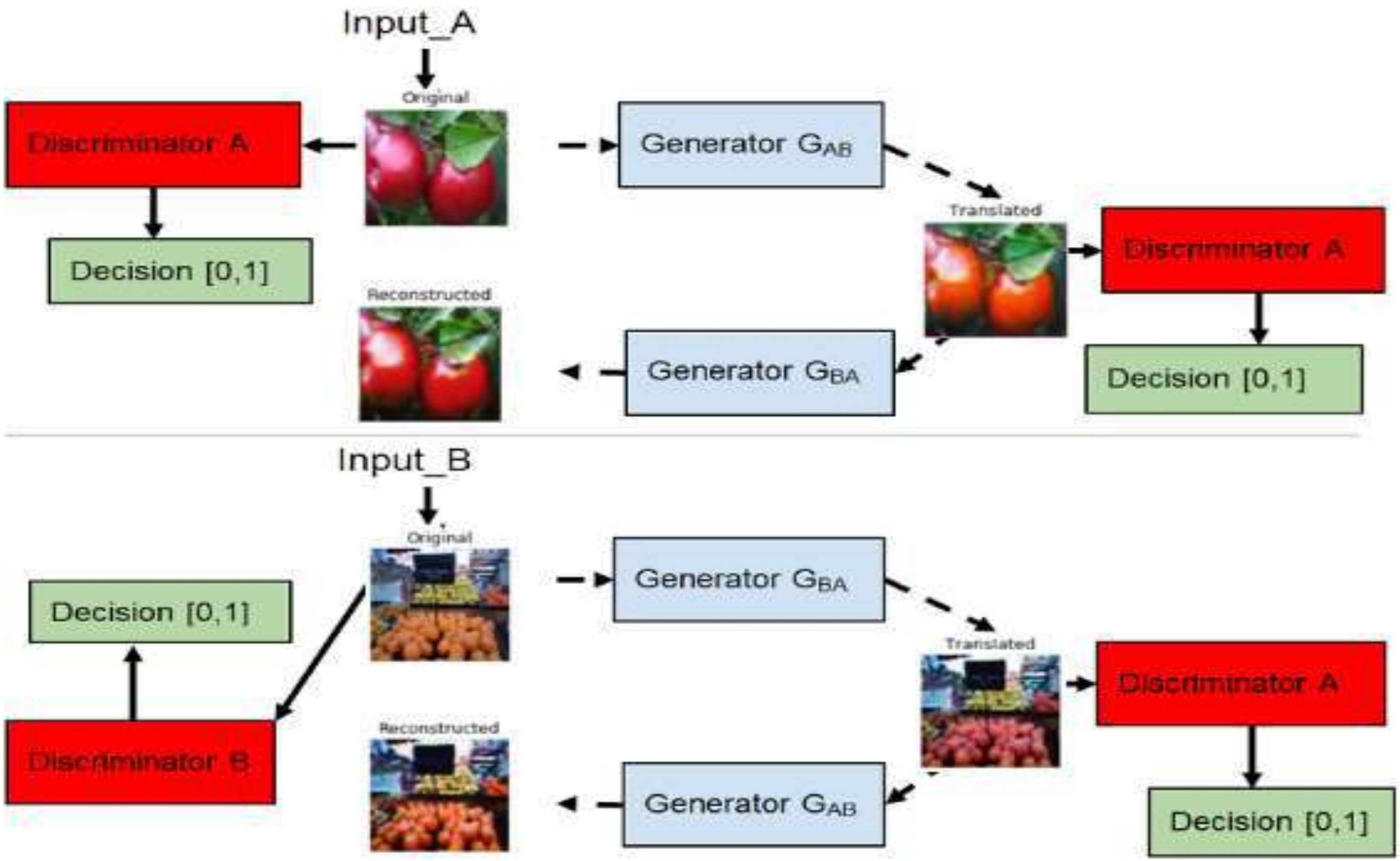
- The most important feature of this cycle_GAN is that it can do this image translation on an unpaired image where there is no relation exists between the input image and output image.





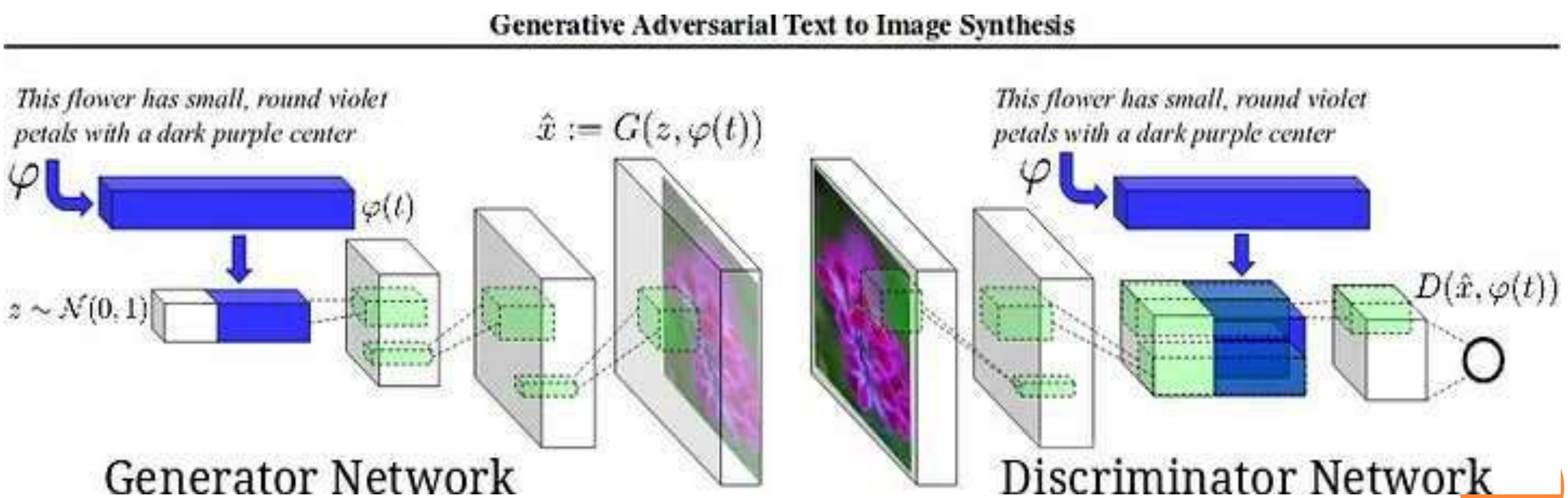
Architecture of Cycle GAN
(CGAN)

CONTD..



Architecture of Cycle GAN
(CGAN)

- **5. Generative Adversarial Text to Image Synthesis:** Text to image synthesis (T2I) is one of the most challenging and interesting tasks in the modern domain of Computer Vision. These GANs can generate images from textual descriptions, bridging the gap between natural language and visual data.

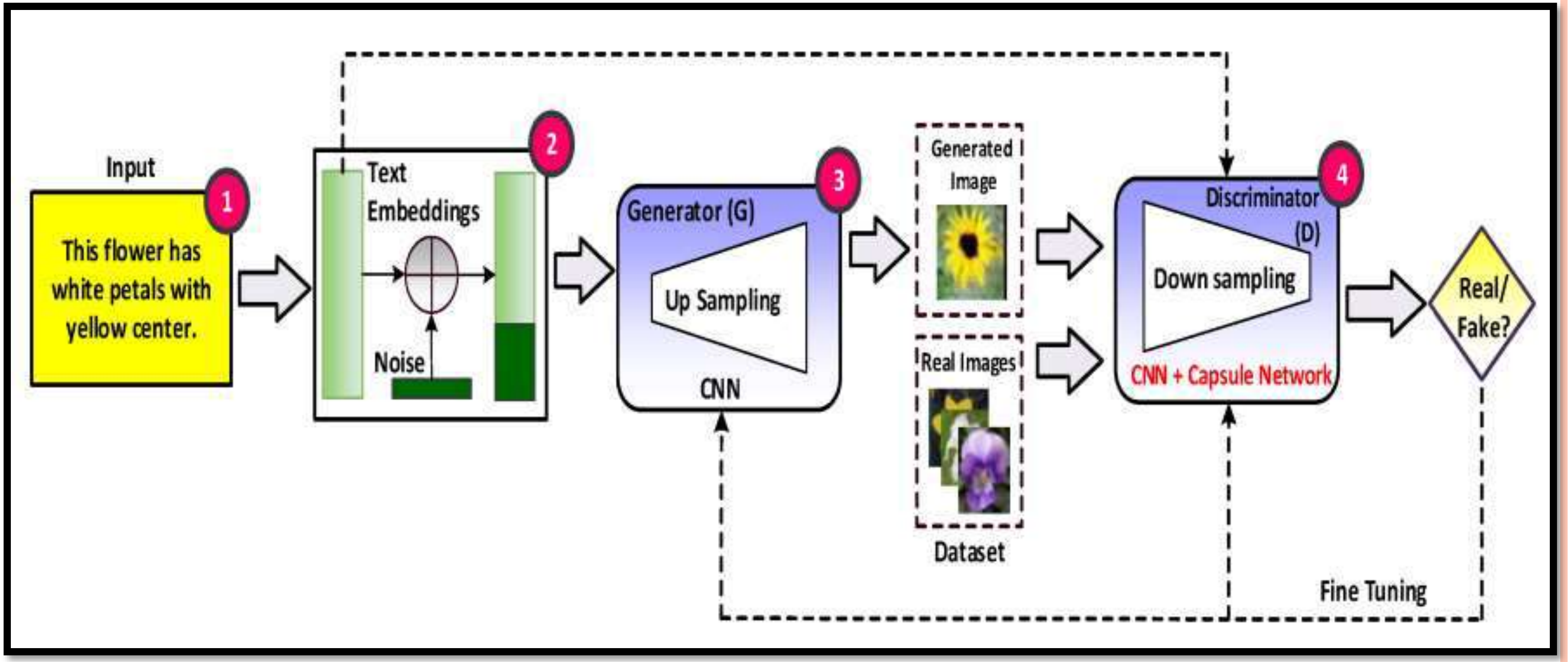


Architecture of Text to Image Synthesis

CONTD..

Example

Contd..

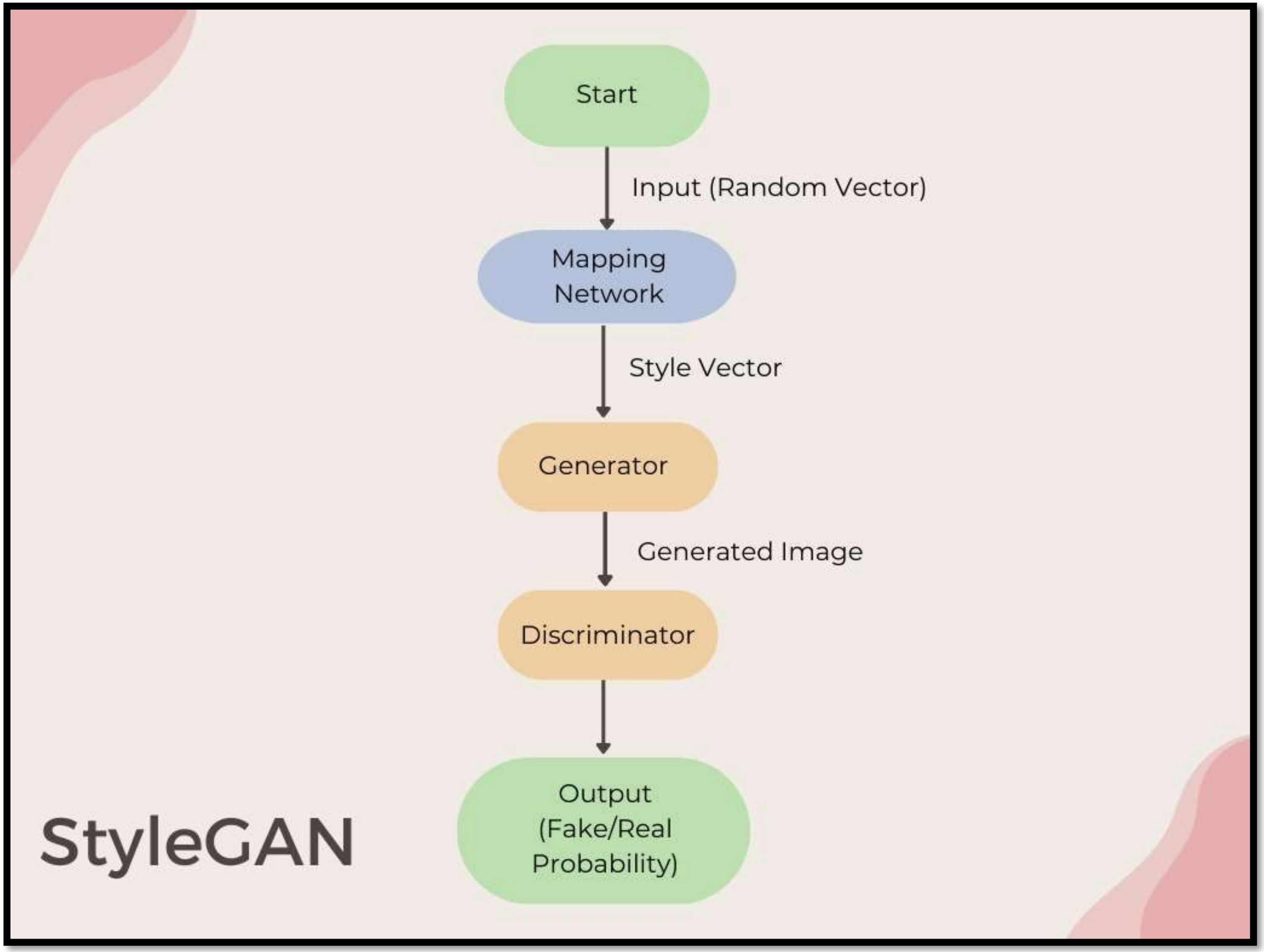


Example

CONTD..

- 6. Style GAN: **Style GAN** proposes a lot of changes in the generator part.
- StyleGAN was designed to create realistic images while manipulating and controlling certain features or styles of the image. These styles associated with the generated images could be **features** like color, texture, pose, etc.

Contd..



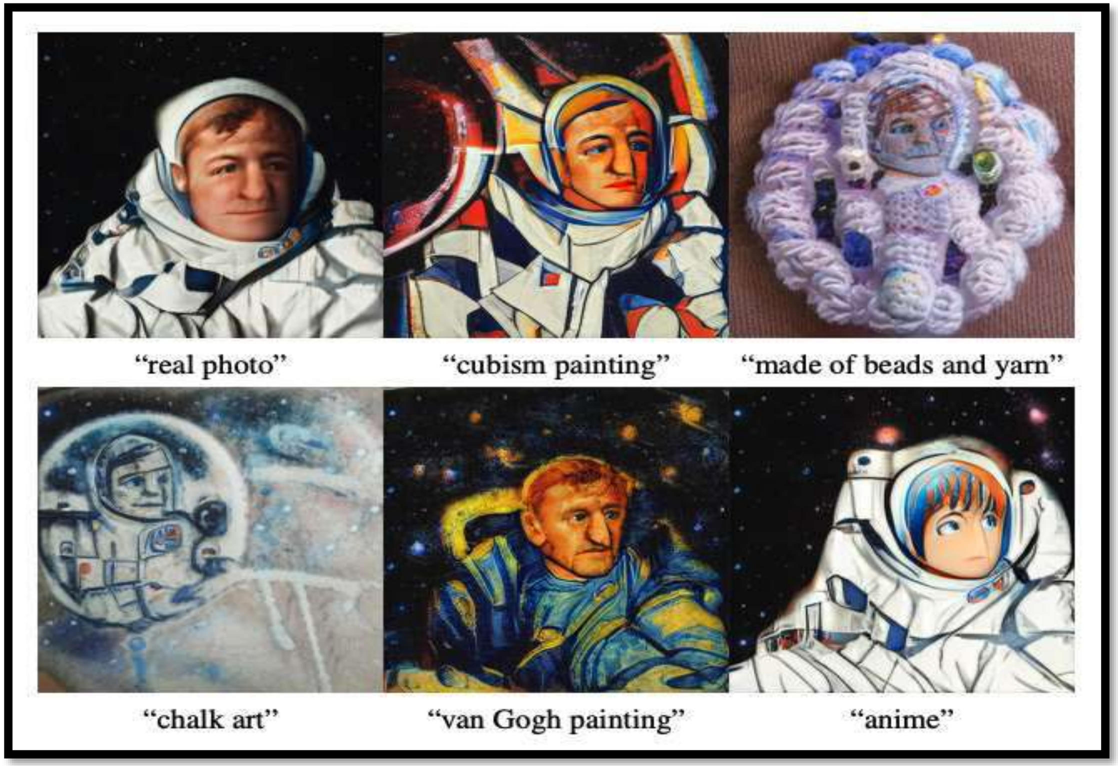
Architecture of Text to Image Synthesis

Contd..

- First, StyleGAN takes a random vector as an input. This vector is **mapped** into a style vector representing different aspects of the image's style and appearance.
- The generator network then generates an image using the style vector.
- The discriminator network evaluates whether the generated image is real or fake (generated).

CONTD..

Content image
Style image
Output image

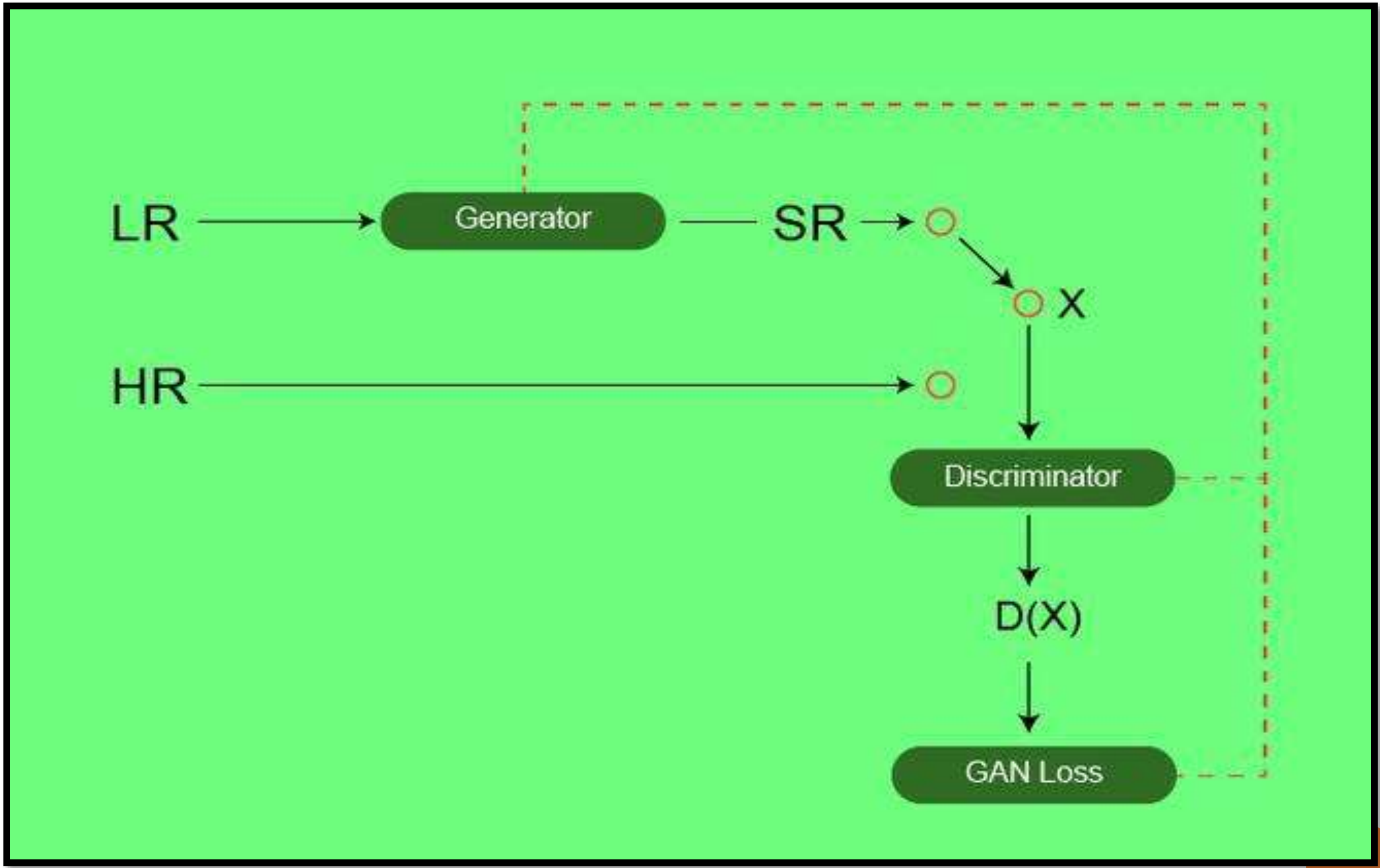


"real photo"
"cubism painting"
"made of beads and yarn"
"chalk art"
"van Gogh painting"
"anime"

Contd..

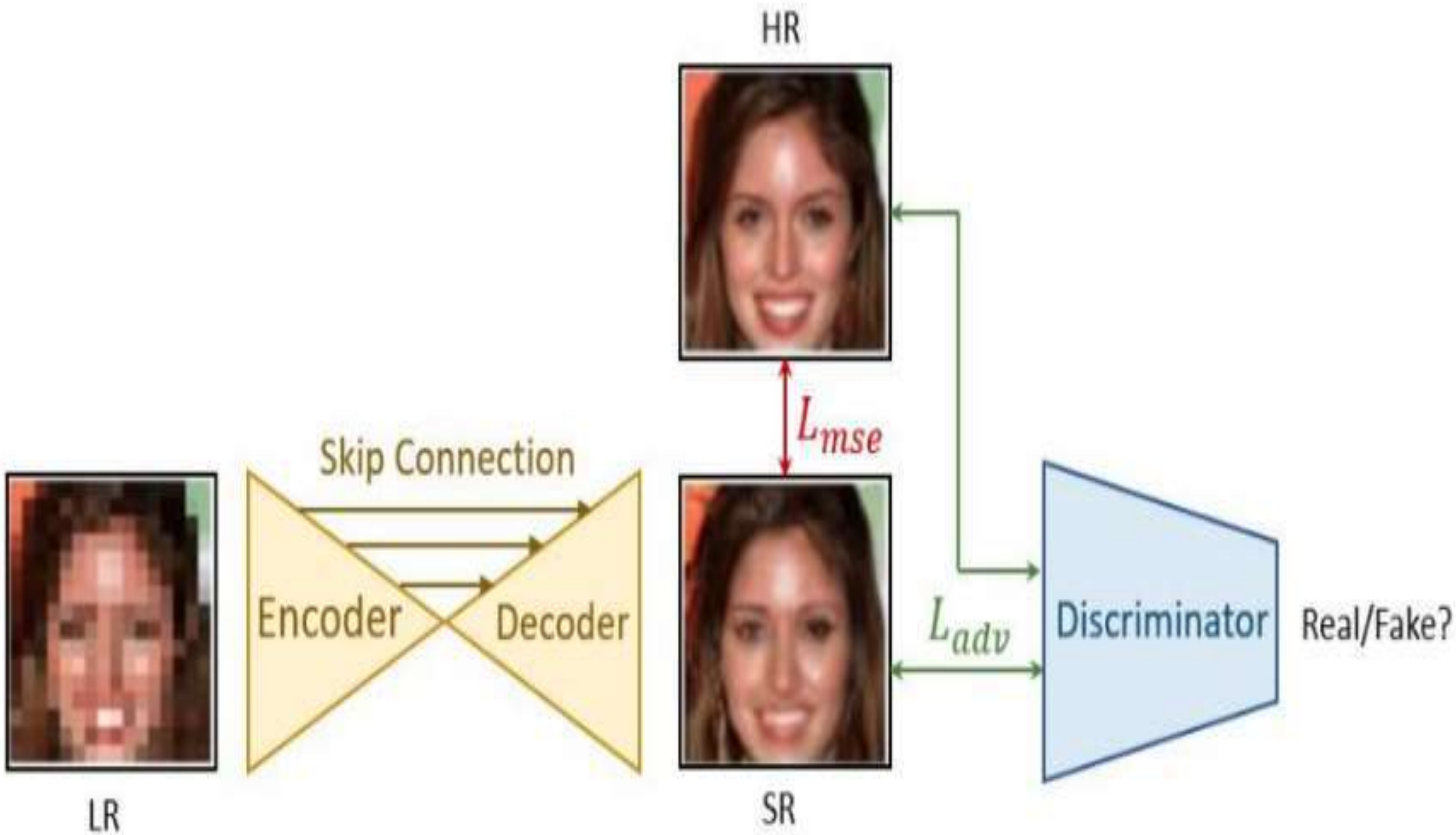
- **7. Super Resolution GAN (SRGAN):** SRGAN enhances the **resolution of images, turning low-resolution inputs into high-resolution outputs without losing detail.**
- SRGAN was proposed by **researchers at Twitter.** The motive of this architecture is to **recover finer textures from the image** when we **upscale it so that it's quality cannot be compromised**.

CONTD..



Architecture of Super Resolution GAN

CONTD..



Example of Super Resolution GAN

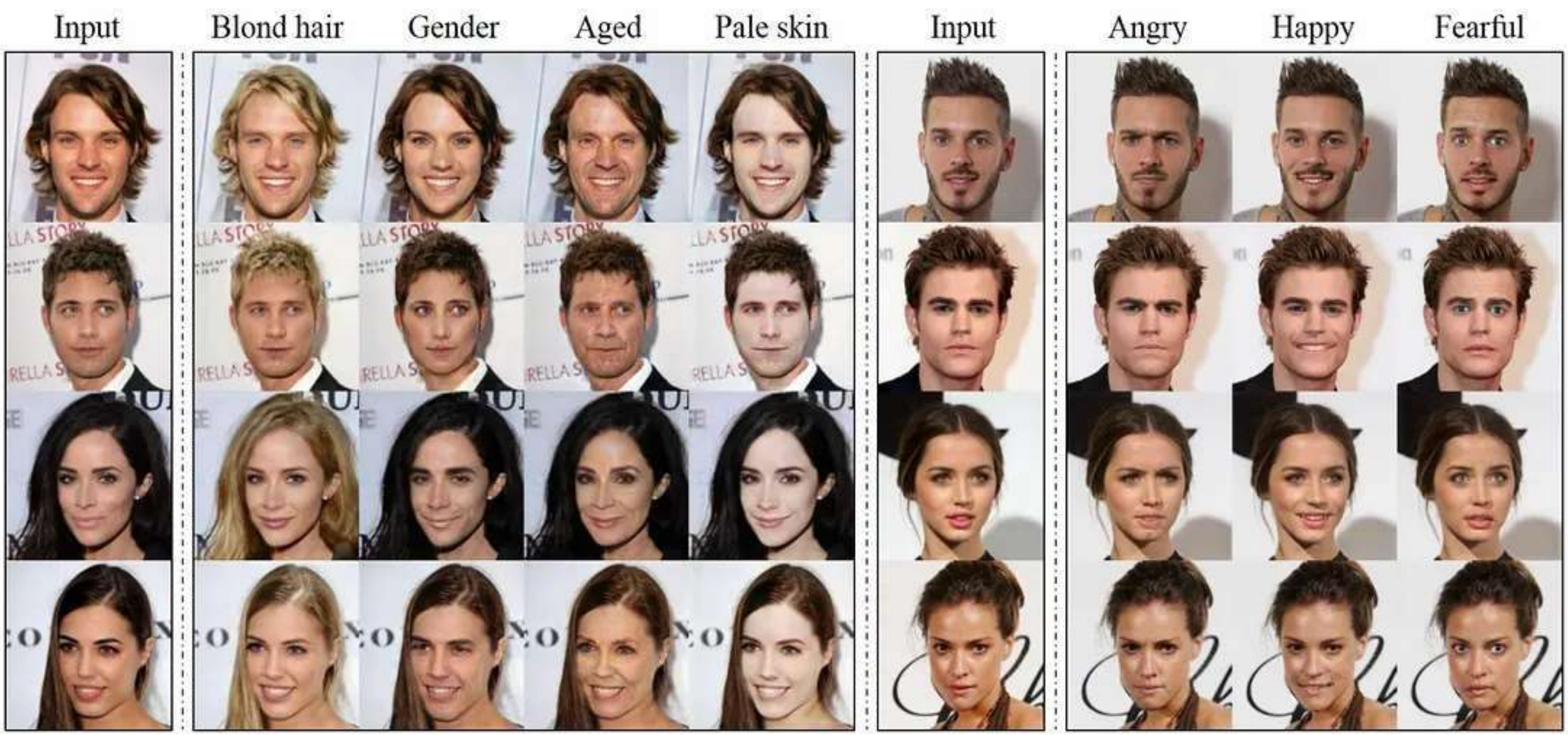
Applications of GAN

- NVIDIA research center has to develop these GAN Applications in real time.
- They generate high quality and photo realistic Images, Videos
- **1. Image to Image Translation:** GANs Can be in use for translating data from images. In image-to-image translations, GANs account for tasks such as:
  - Changing sketches to **color photographs.**
  - Converting satellite images to google maps.
  - Translation of photos from day to night and vice versa.
  - **Translation of black and white photographs to color.**

Contd...

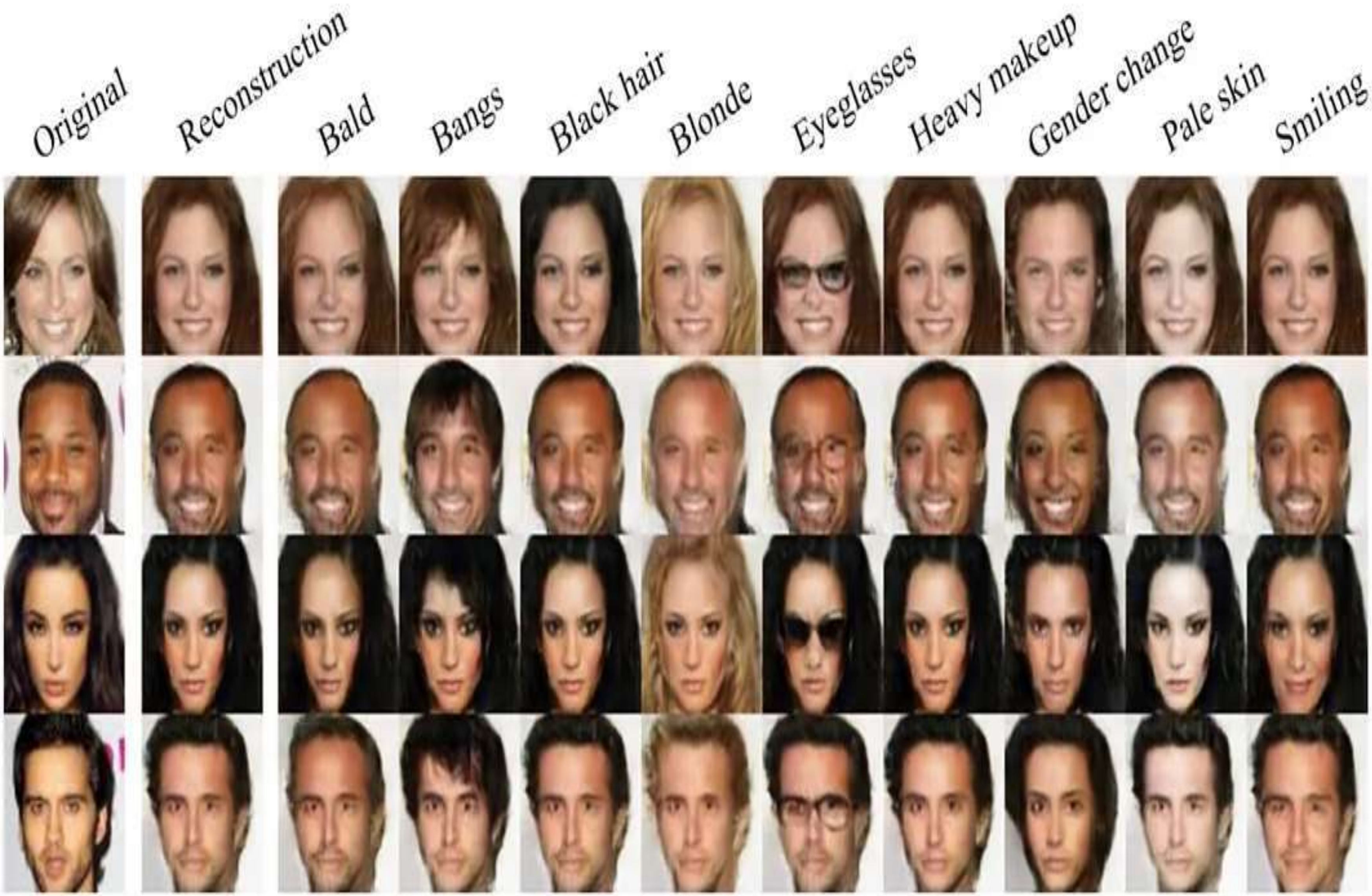
CONTD..

- StarGAN is an image-to-image translation for one domain to another. For example, given a happy face, we want to transform it into a fearful face.

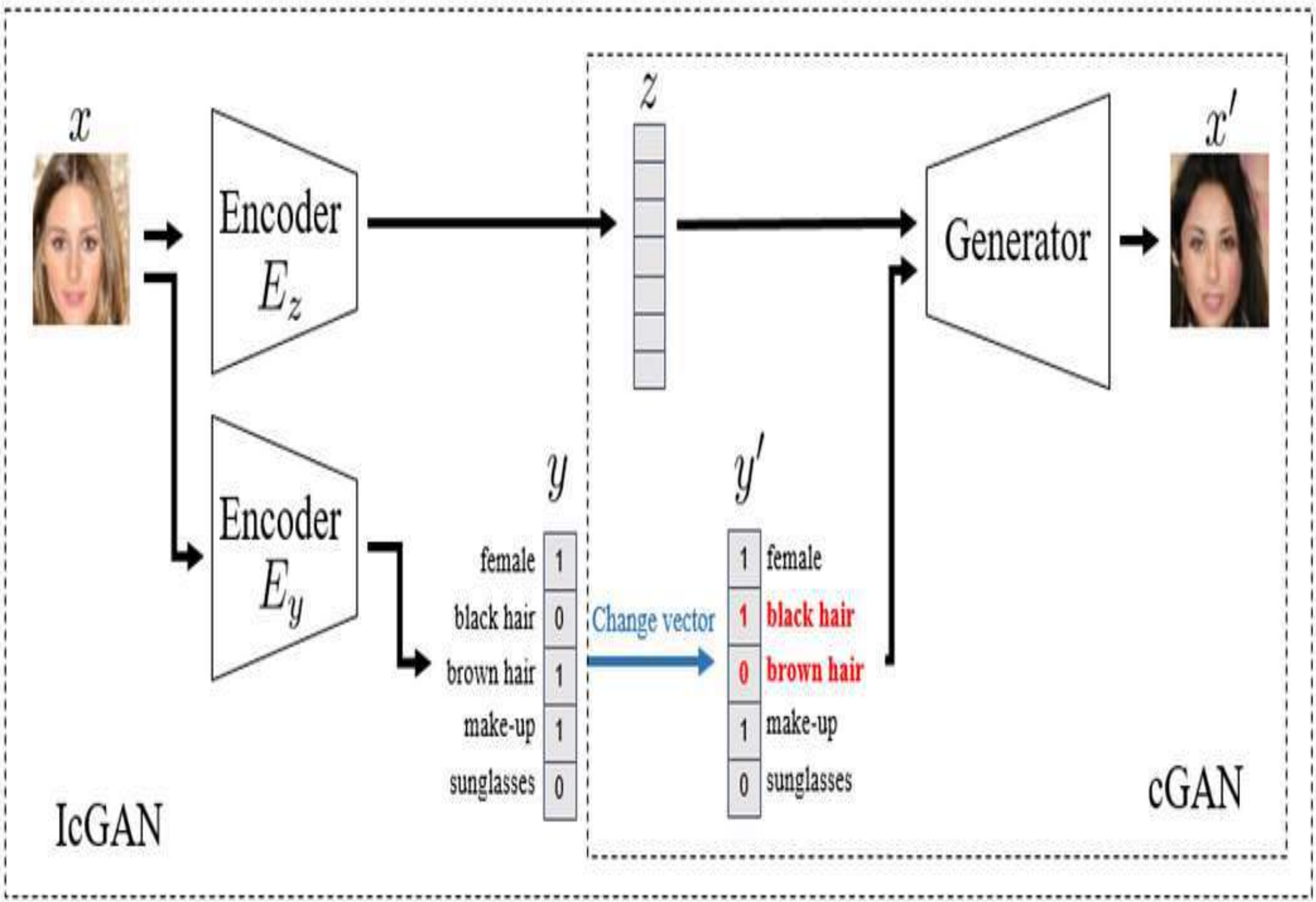


- **2. Image Editing:** Most image editing software **these days don't give us much flexibility** to **make creative changes in pictures.**
- **For example,** let's say you want to change the **appearance of a 90-year-old person by changing his/her hairstyle**. This can't be done by the current image editing tools out there. But guess what? Using GANs, **we can reconstruct images and attempt to change the appearance drastically**.

CONTD..


Original
Reconstruction
Bald
Bangs
Black hair
Blonde
Eyeglasses
Heavy makeup
Gender change
Pale skin
Smiling

CONTD..

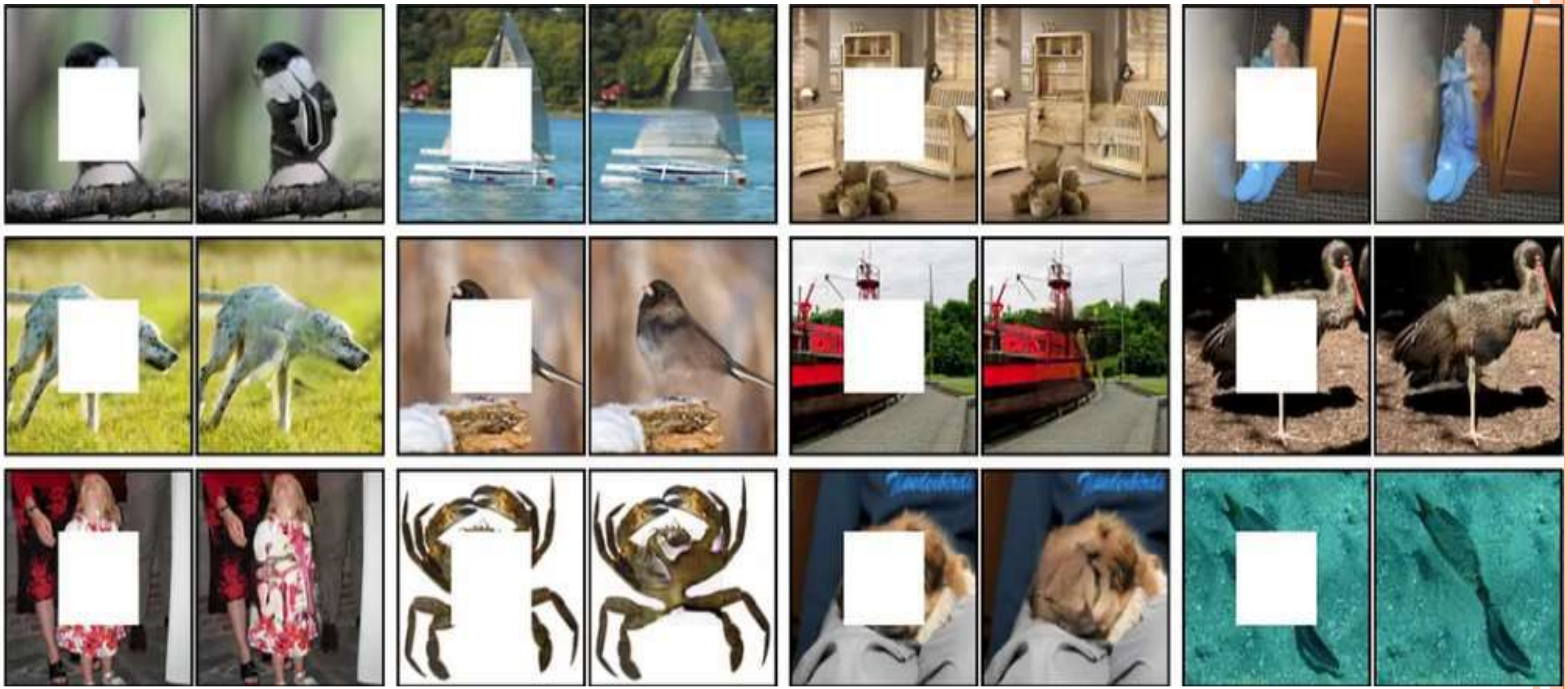


x
Encoder
E_z
z
Generator
x'
Encoder
E_y
y
female 1
black hair 0
brown hair 1
make-up 1
sunglasses 0
Change vector
y'
1 female
1 black hair
0 brown hair
1 make-up
0 sunglasses
cGAN
IcGAN

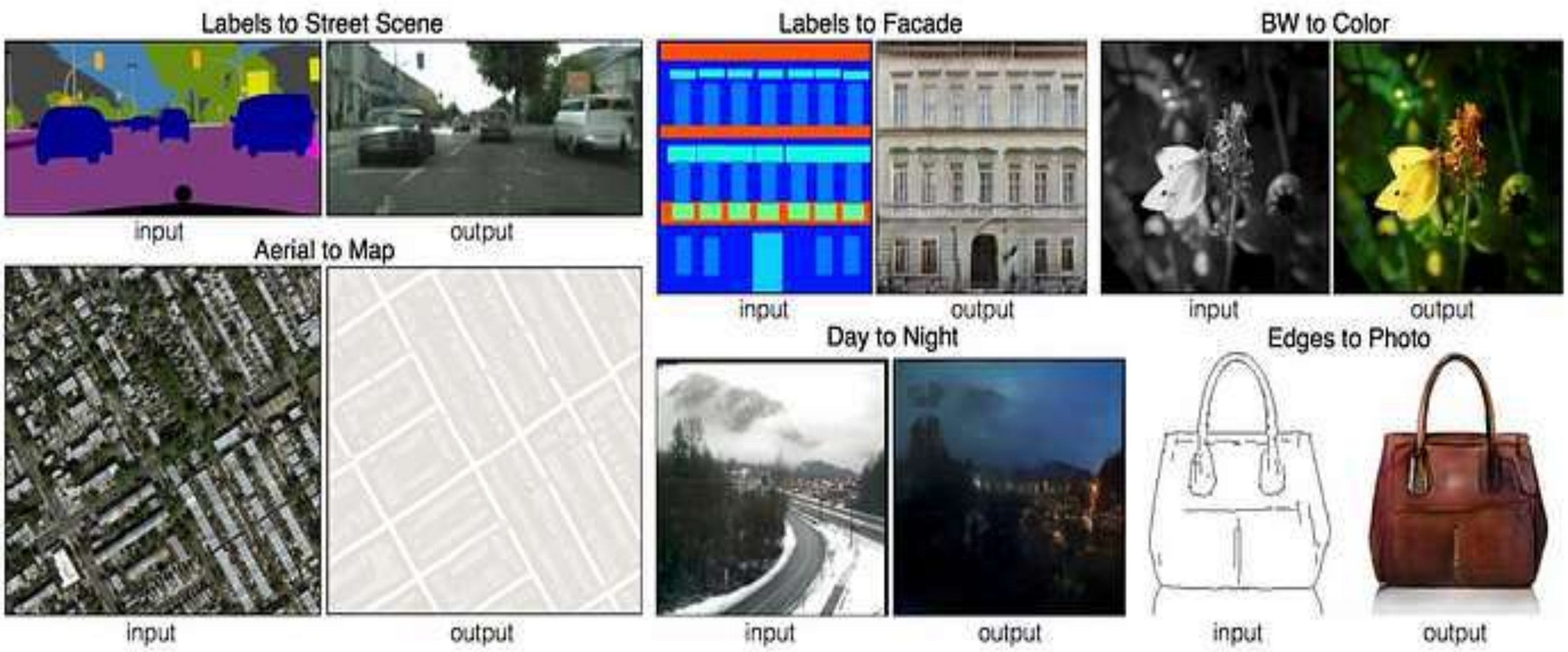
- **4. Face Synthesis:** Synthesis faces in different poses: With a single input image, we create faces in different viewing angles.
- For example, we can use this to transform images that will be easier for face recognition.

. Synthesis results under various illuminations. The first row is the synthesized image, the second row is the input.

- <u>**5. Image Painting:**</u> **Repair images** have been an important subject decades ago. **GAN is used to repair images** and **fill the missing part** with **created "content".**



- **6. Pix to Pix:** Pix2Pix is an **image-to-image translation** that get quoted in **cross-domain GAN**'s frequently. **For example**, **it converts a satellite image into a map (the bottom left).**



- **DeblurGAN** performs **motion deblurring.**

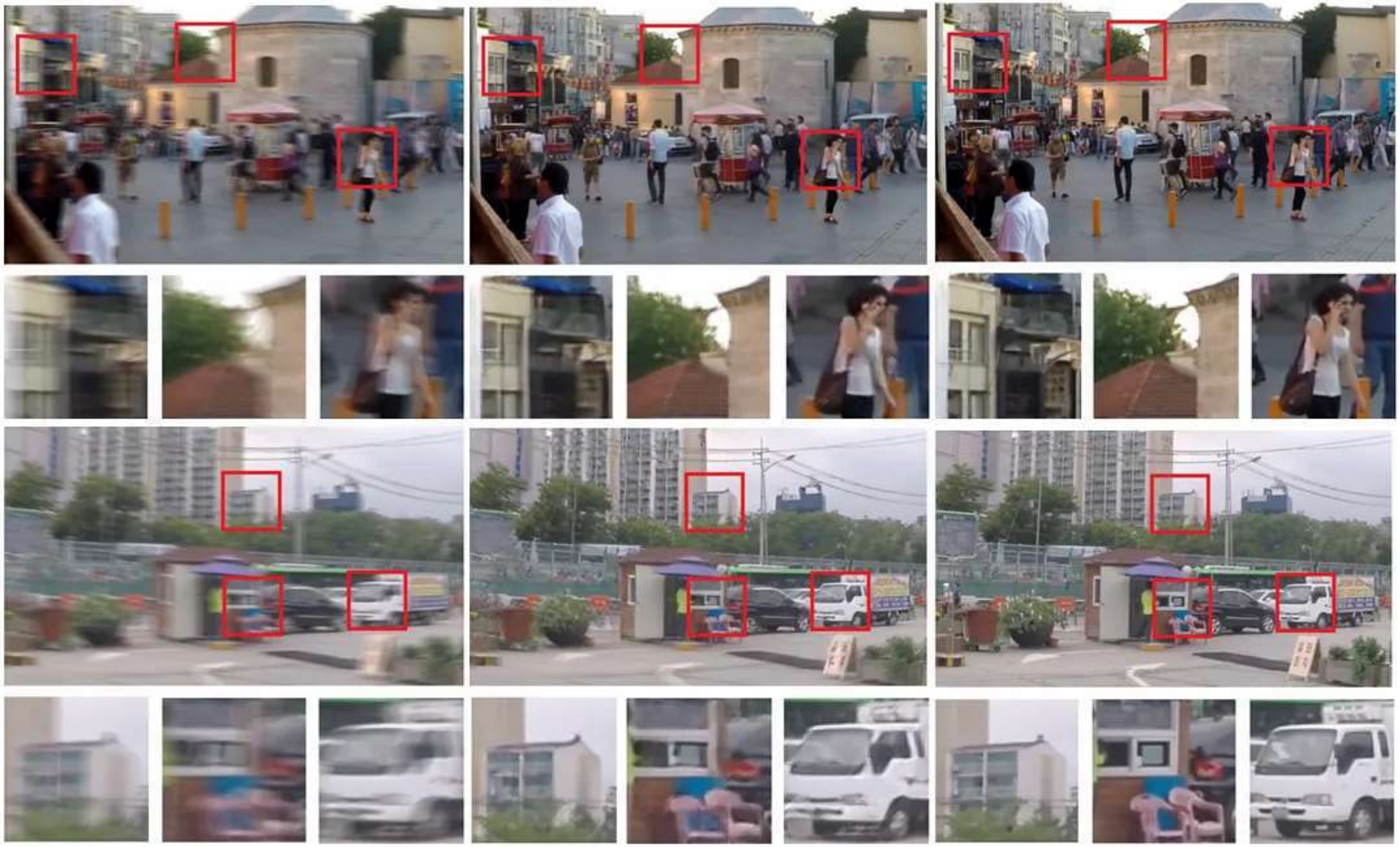


Figure 2: GoPro images [25] processed by DeblurGAN. Blurred – left, DeblurGAN – center, ground truth sharp – right.

- **7.Music Generation:** GAN can be applied to non-image domain, like composing music.

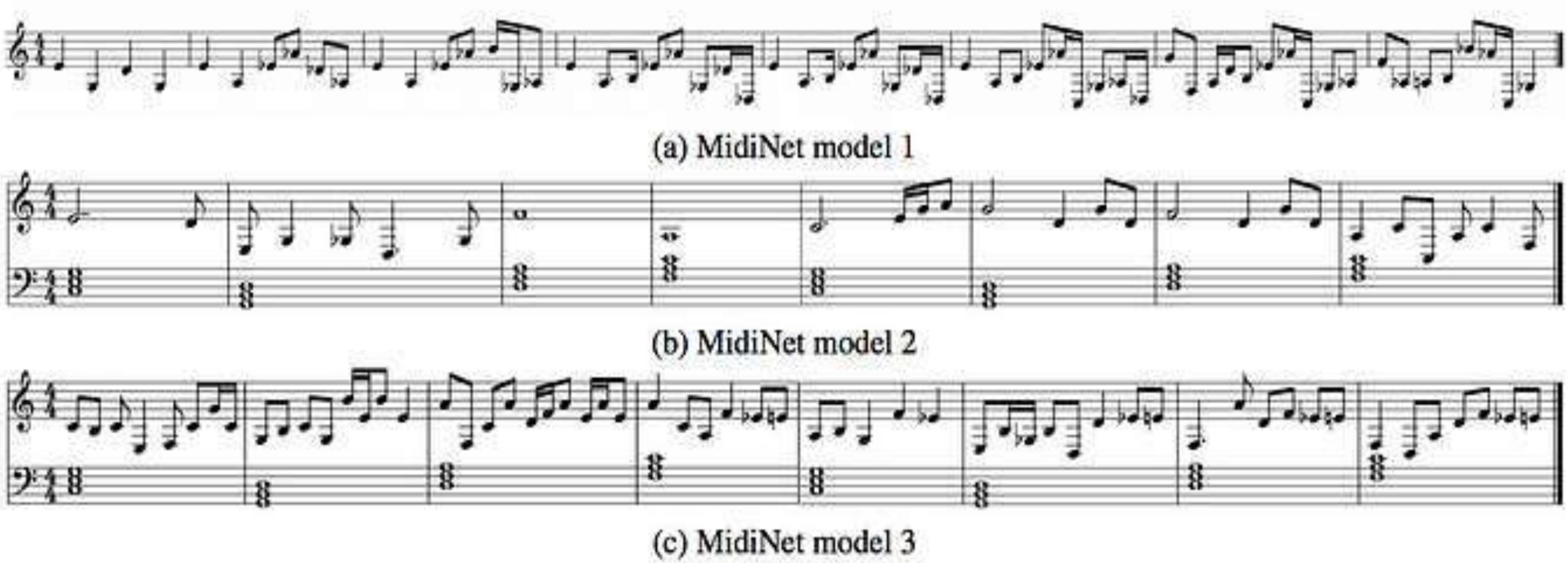
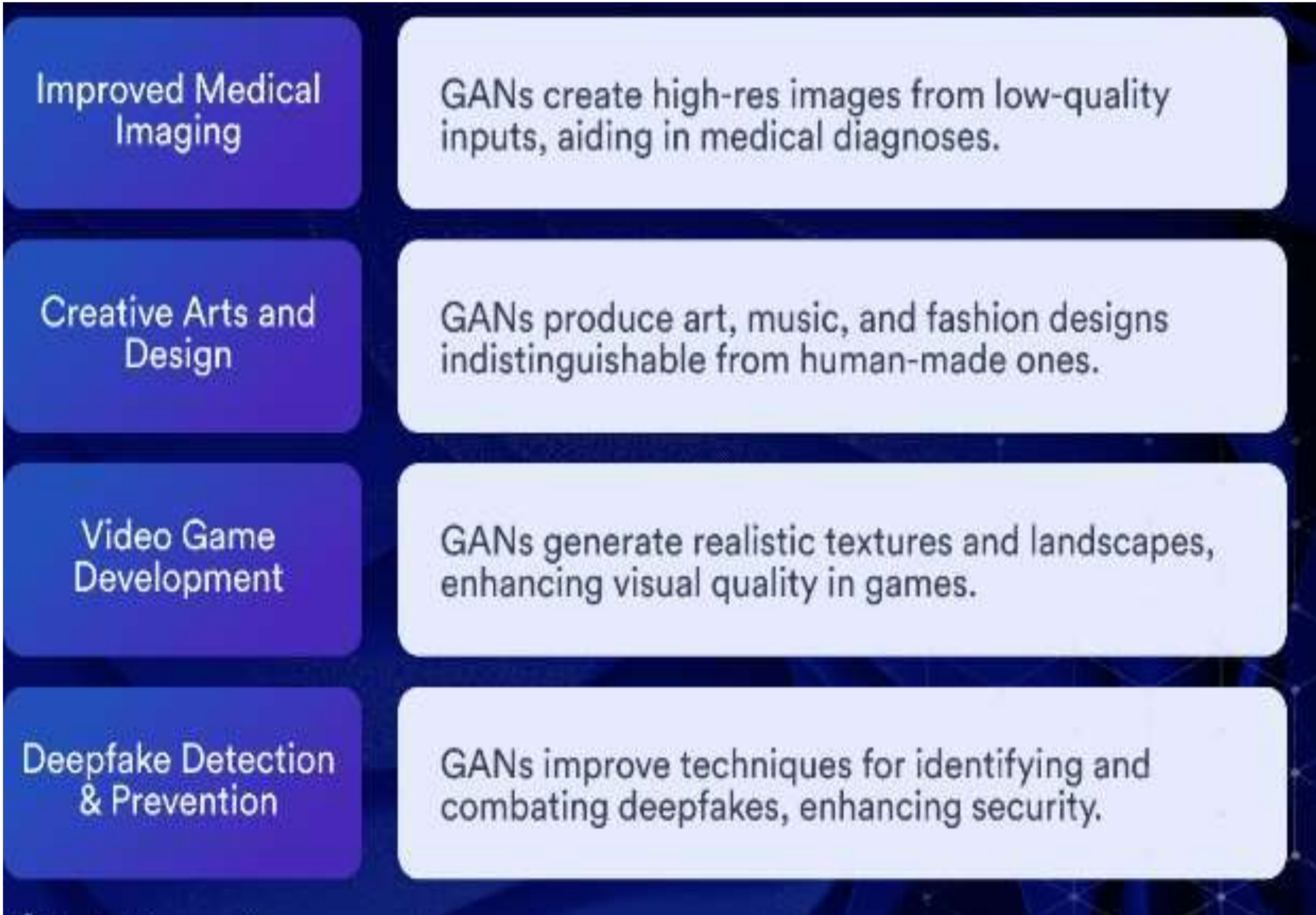
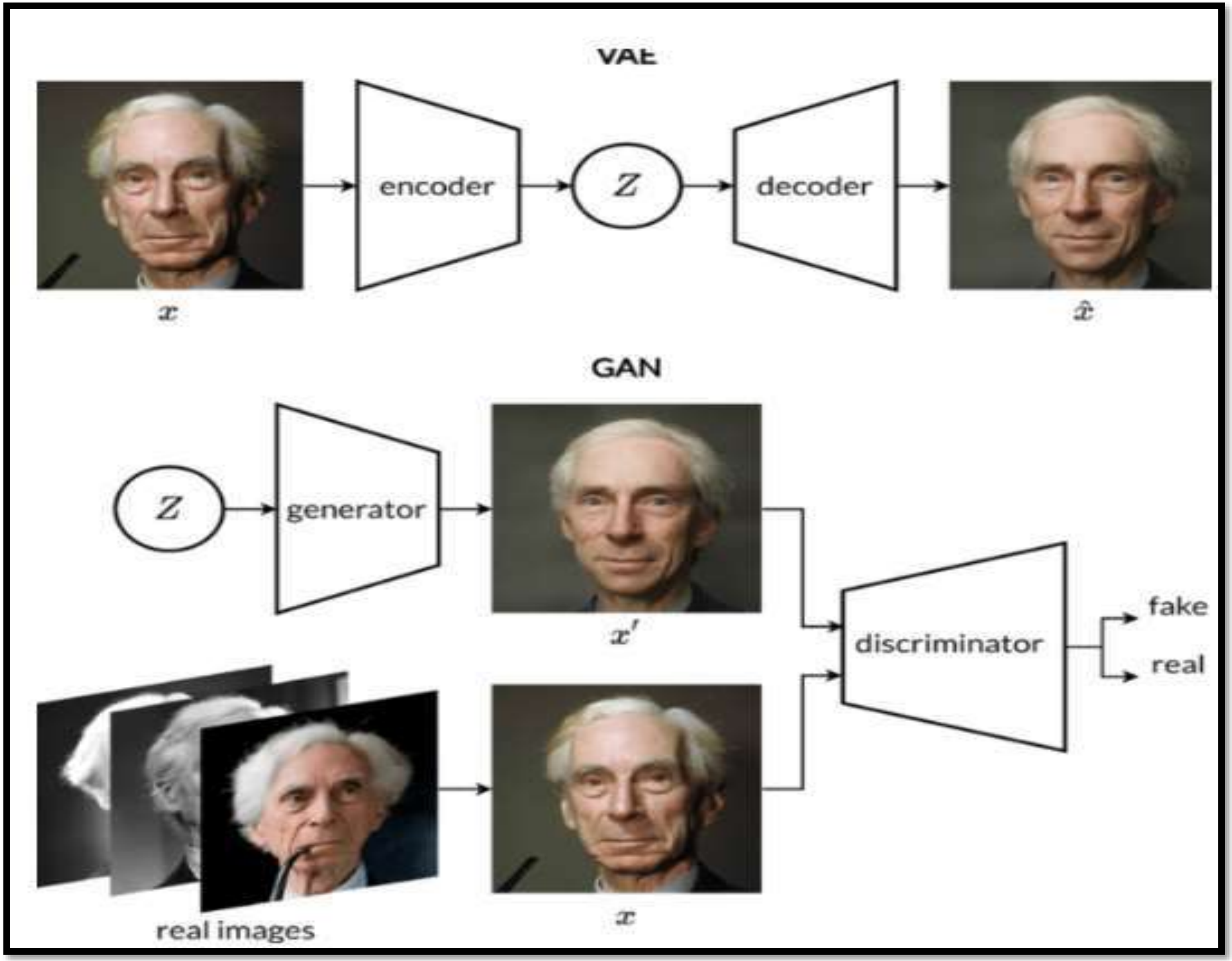


Figure 3. Example result of the melodies (of 8 bars) generated by different implementations of MidiNet.

Future Generations of GANs



DIFFERENCES B/W VAN AND GAN



CONTD..

Topics	Generative Adversarial Networks	Variational Autoencoder
Functionality	Composed of two models (**a generator and a discriminator**) that compete with each other. The **generator creates fake samples** and the discriminator attempts to distinguish between **real and fake samples.**	Composed of an **encoder and a decoder**. The **encoder maps inputs to a latent space,** and the **decoder maps points in the latent space back to the input space.**
Output Quality	Can **generate high-quality, realistic outputs.** Known for **generating images that are hard to distinguish from real ones.**	Generally produces **less sharp or slightly blurrier images compared to GANs**. However, this may depend on the specific implementation and problem domain.
Training Stability	Training GANs can be **challenging and unstable,** due to the **adversarial loss used in training.**	Generally **easier and more stable to train** because they use a **likelihood-based objective function.**

Thank You