# Generative Adversarial Networks (GANs)

# What is a GAN?

**Definition:**

A GAN is a machine learning framework designed to generate new data samples that resemble a given dataset.

It involves two neural networks: a Generator and a Discriminator, trained simultaneously.

**Key Concept:**

The adversarial process where the Generator creates data and the Discriminator evaluates it.

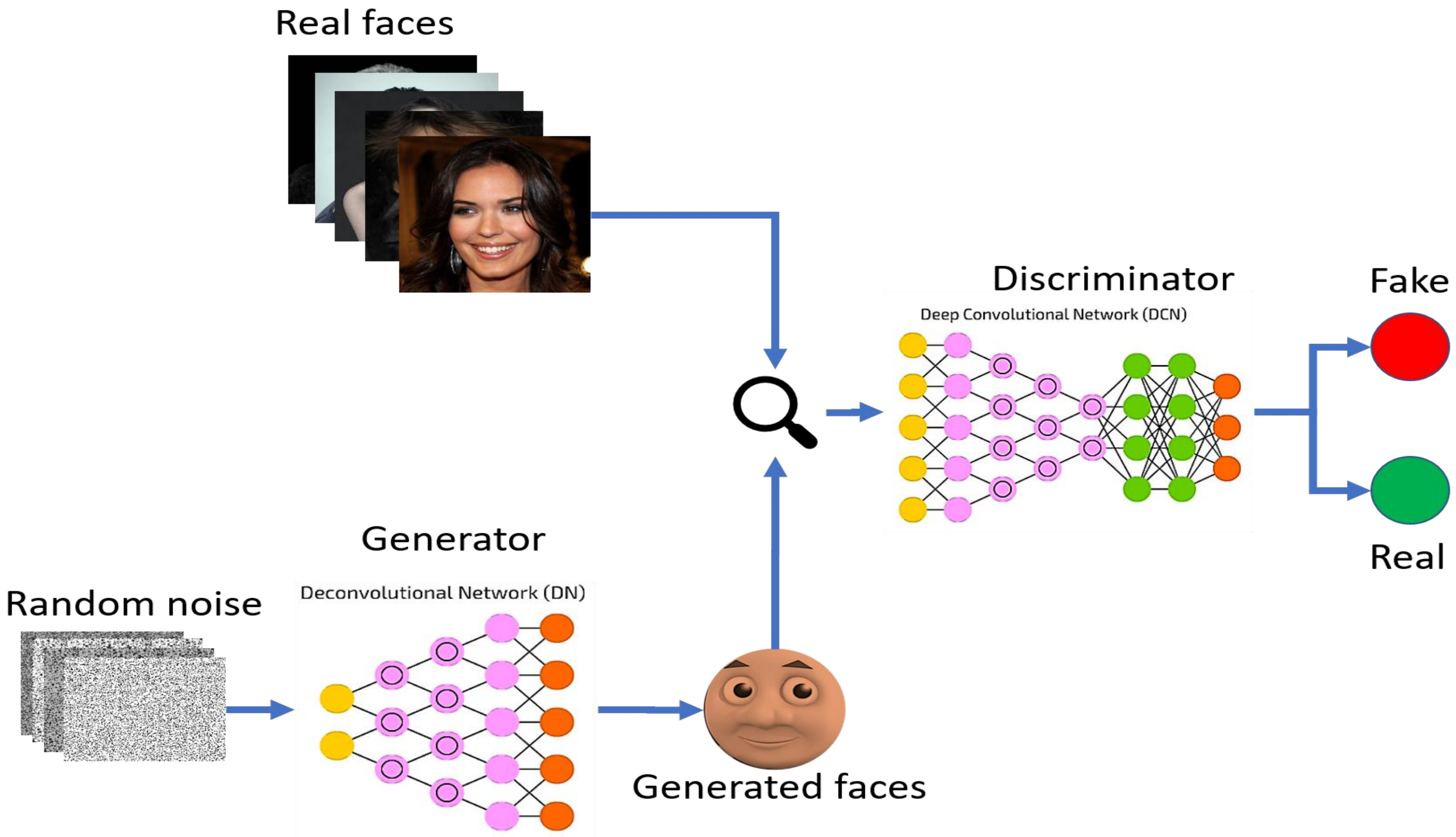
# Components of GAN

## Generator (G):

**Purpose:** Generate synthetic data samples.
**Input:** Random noise (typically a vector of random numbers).
**Output:** Data sample (e.g., image, text, etc.).
**Goal:** Create data indistinguishable from real data to fool the Discriminator.

## Discriminator (D):

**Purpose:** Distinguish between real data samples (from the training set) and fake samples (from the Generator).
**Input:** A data sample.
**Output:** Probability score (real or fake).
**Goal:** Accurately identify real versus generated data.


Real faces
Discriminator
Deep Convolutional Network (DCN)
Fake
Real
Generator
Random noise
Deconvolutional Network (DN)
Generated faces

# How GANs Work

**Training Process:**

- The Generator takes random noise and generates a data sample.
- This generated sample, along with real samples from the dataset, is fed into the Discriminator.
- The Discriminator evaluates the samples and classifies them as real or fake.
- Both networks are updated based on the Discriminator's feedback.

**Adversarial Training:**

- The Generator improves by learning to produce more realistic data to fool the Discriminator.
- The Discriminator enhances its ability to differentiate real data from fake data.
- The two networks engage in a zero-sum game, where the improvement of one network depends on the performance of the other.

# Loss Function

ground truth $= y$, prediction $= \hat{y}$

Mean Squared Error Loss

$$L = \frac{1}{m}\sum(y_i - \hat{y_i})^2$$

$$L = (y - \hat{y})^2$$

Binary Cross-entropy Loss

$$L = -\frac{1}{m}\sum[y_i ln(\hat{y_i}) + (1 - y_i) ln(1 - \hat{y_i})]$$

$$L = y ln(\hat{y}) + (1 - y) ln(1 - \hat{y})$$

# Loss Functions

**Discriminator Loss (D Loss):**
Measures the Discriminator's ability to distinguish real data from fake data.

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

The goal is to maximize the probability of correctly classifying real and fake samples.

**Generator Loss (G Loss):**

Measures the Generator's success in fooling the Discriminator.

$$\mathcal{L}_G = -\mathbb{E}_{z\sim p_z(z)}[\log D(G(z))]$$

The goal is to minimize this loss, which means generating samples that the Discriminator considers as real.

# Objective Function

$$\max_{\theta} E_{x \sim p_{data}}[\log D_\theta(x)] + E_{z \sim p(z)}[\log(1 - D_\theta(G_\phi(z)))]$$

# Applications of GANs

**Image Generation:**

- GANs can create realistic images from random noise.
- Examples include generating faces, landscapes, and other objects.

**Image-to-Image Translation:**

- Converts images from one domain to another (e.g., sketches to photorealistic images).
- Applications in style transfer and data augmentation.

**Super-Resolution:**

- Enhances the resolution of low-quality images.
- Used in medical imaging and satellite image processing.

**Data Augmentation:**
Generates additional data samples to augment training datasets.
Useful in scenarios with limited data availability.

**Text-to-Image Synthesis:**
Creates images based on textual descriptions.
Useful in creative industries and for generating visual content from written input.

# Summary

**Key Points:**

GANs consist of two neural networks (Generator and Discriminator) trained in an adversarial setting.

The Generator aims to create realistic data, while the Discriminator aims to differentiate real from fake data.

GANs have numerous applications in image generation, data augmentation, and more.
Despite their potential, GANs face challenges such as training instability and mode collapse.

# Thank You