MINOR PROJECT

Project Name- Image Magick
Semester- 6th A2

Team Members:
- **Arjun Kapoor (2021a1r061) –[LEADER]**
- **Madhu Bala (2021a1r077)**
- **Manmeet Kour (2021a1r090)**

श्रेष्ठ

Model Institute of
Engineering & Technology

Team Members

Leader

Group member

Group member

Table Contents

Project Objective and Outcome

Objective:- The goal of this project is to use Machine Learning to transform police sketches to realistic images.

Outcome:- The project successfully utilized Machine Learning techniques to transform police sketches into realistic images. By training models on a dataset of police sketches and matching real photos, the system learned to create detailed, lifelike images from simple sketches. Now , it can produce realistic pictures from basic sketches accurately.

Problem Statement:

The goal of this project is to develop a robust Machine Learning system that can convert police sketches into realistic images, thereby enhancing the effectiveness of sketches in criminal investigations. This system should be capable of accurately translating the key features of a sketch into a lifelike image that can be reliably used to identify suspects, thus improving the overall efficiency and accuracy of law enforcement operations.

Problem Solution:

This project proposes to develop an ML model that can transform basic police sketches into realistic images. The model will be trained on a dataset of paired images: police sketches alongside corresponding real photographs of people. By learning the relationships between facial features in sketches and their realistic counterparts, the model will be able to generate more detailed and accurate images that can aid in suspect identification.

Technology Stack

- **Language Used:** Python
- **Algorithms Used:** GAN (Generative adversarial Network)
- **Libraries Used:** NumPy, TensorFlow and Keras

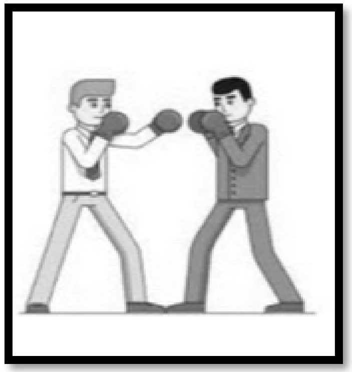
How GAN Work?

Generative Adversarial Networks (GANs) can be broken down into three parts:

•**Generative:** Generates data (Creates fake data).

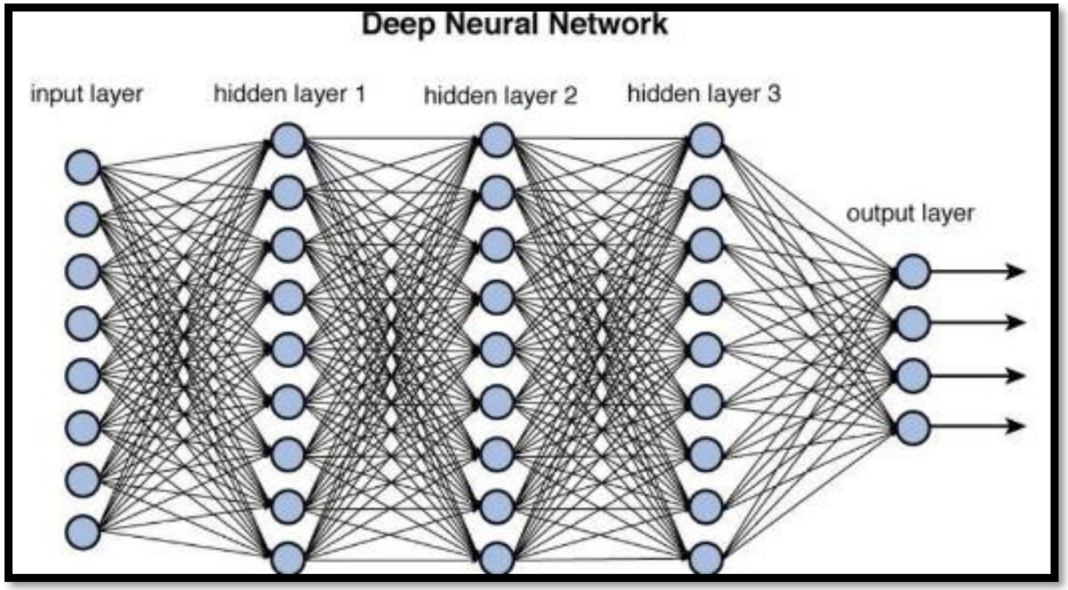
In a generative adversarial network (GAN), "generative" refers to a class of statistical models that can create new data instances.

•**Adversarial:** Generator and discriminator, each competing to win, **Generator** trying to fake and **Discriminator** , trying not to be fooled .



In Generative Adversarial Networks (GANs), the term "adversarial" refers to the training environment for the two neural networks that make up the GAN.

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



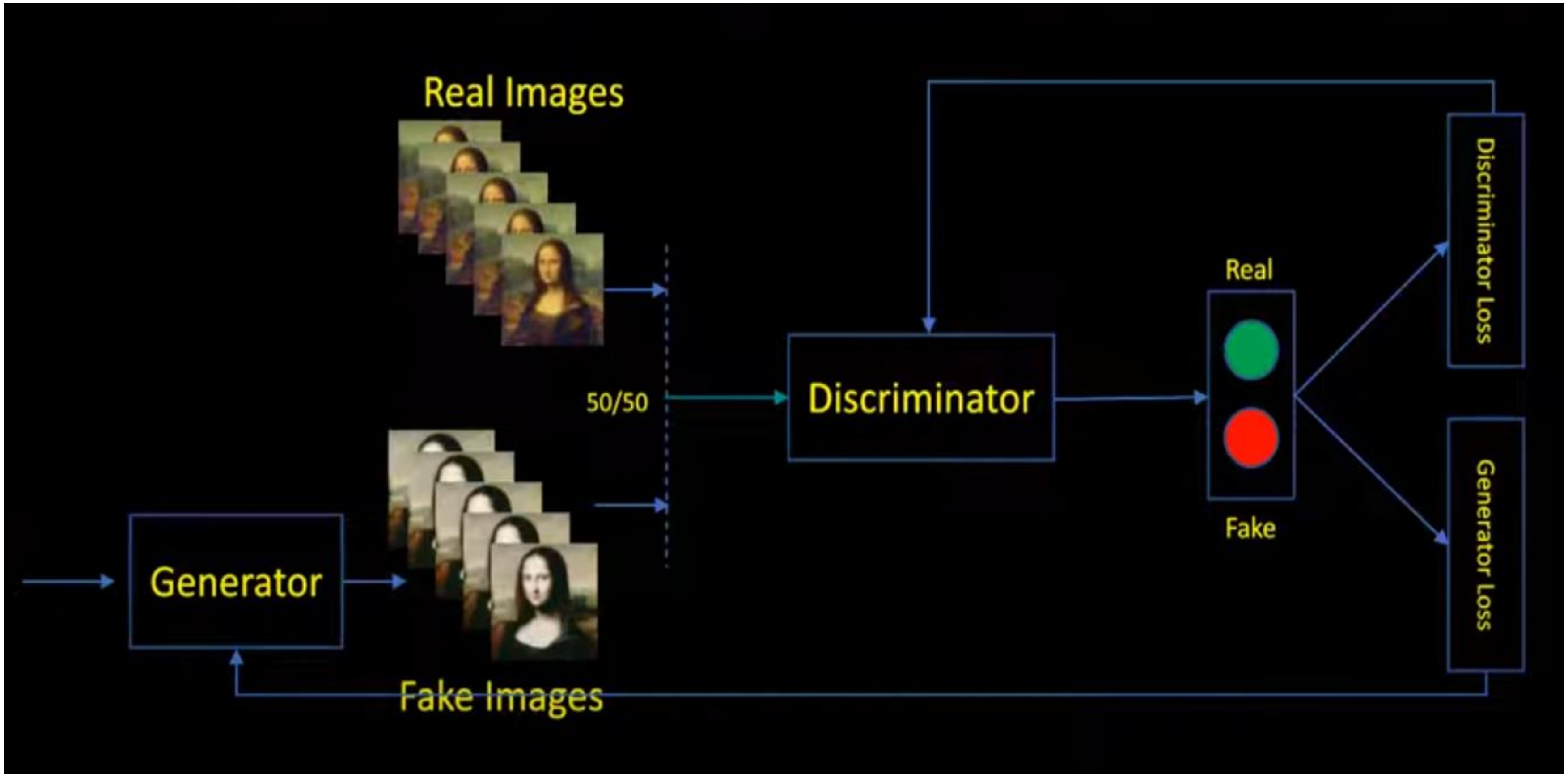


Figure: GAN (Generative Adversarial Network)

A **generative adversarial network (GAN)** has two parts:

The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator. Used to generate new images which look like real images.

The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results. Used to classify images as real or fake.

Discriminator Training Data:

The discriminator's training data comes from two sources:

Real data instances, such as real pictures of people. The discriminator uses these instances as positive examples during training.

Fake data instances created by the generator. The discriminator uses these instances as negative examples during training.

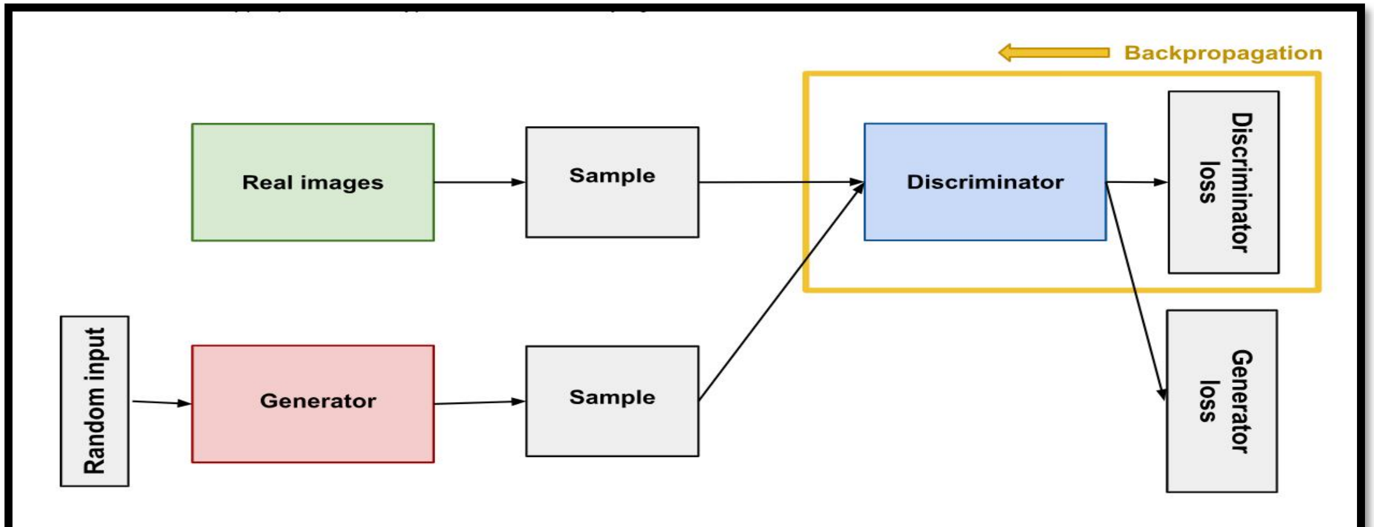


Figure 1: Backpropagation in discriminator training.

The Generator:

The generator part of a GAN learns to create fake data by incorporating feedback from the discriminator. It learns to make the discriminator classify its output as real.

Generator training requires tighter integration between the generator and the discriminator than discriminator training requires. The portion of the GAN that trains the generator includes:

•random input

•generator network, which transforms the random input into a data instance

•discriminator network, which classifies the generated data

•discriminator output

•generator loss, which penalizes the generator for failing to fool the discriminator.

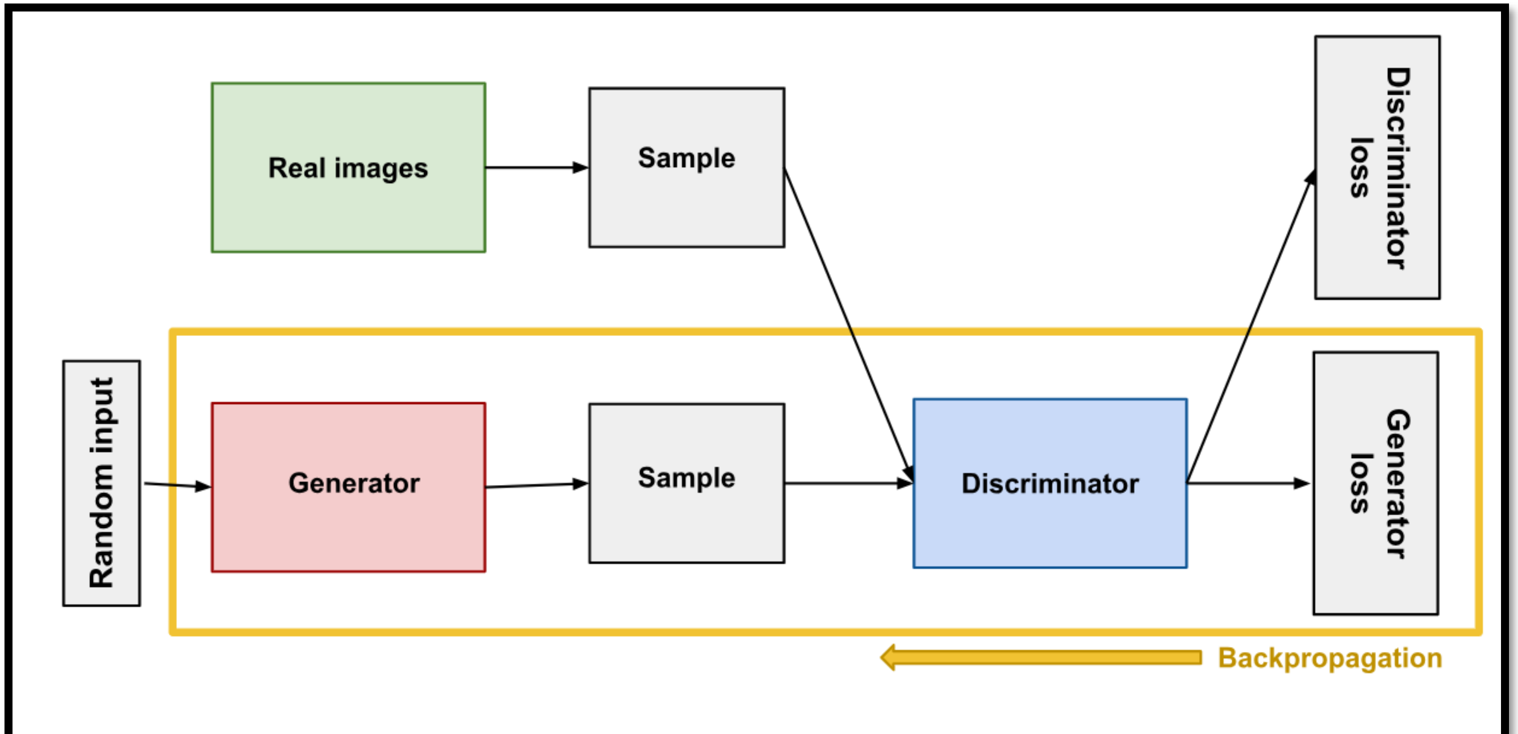
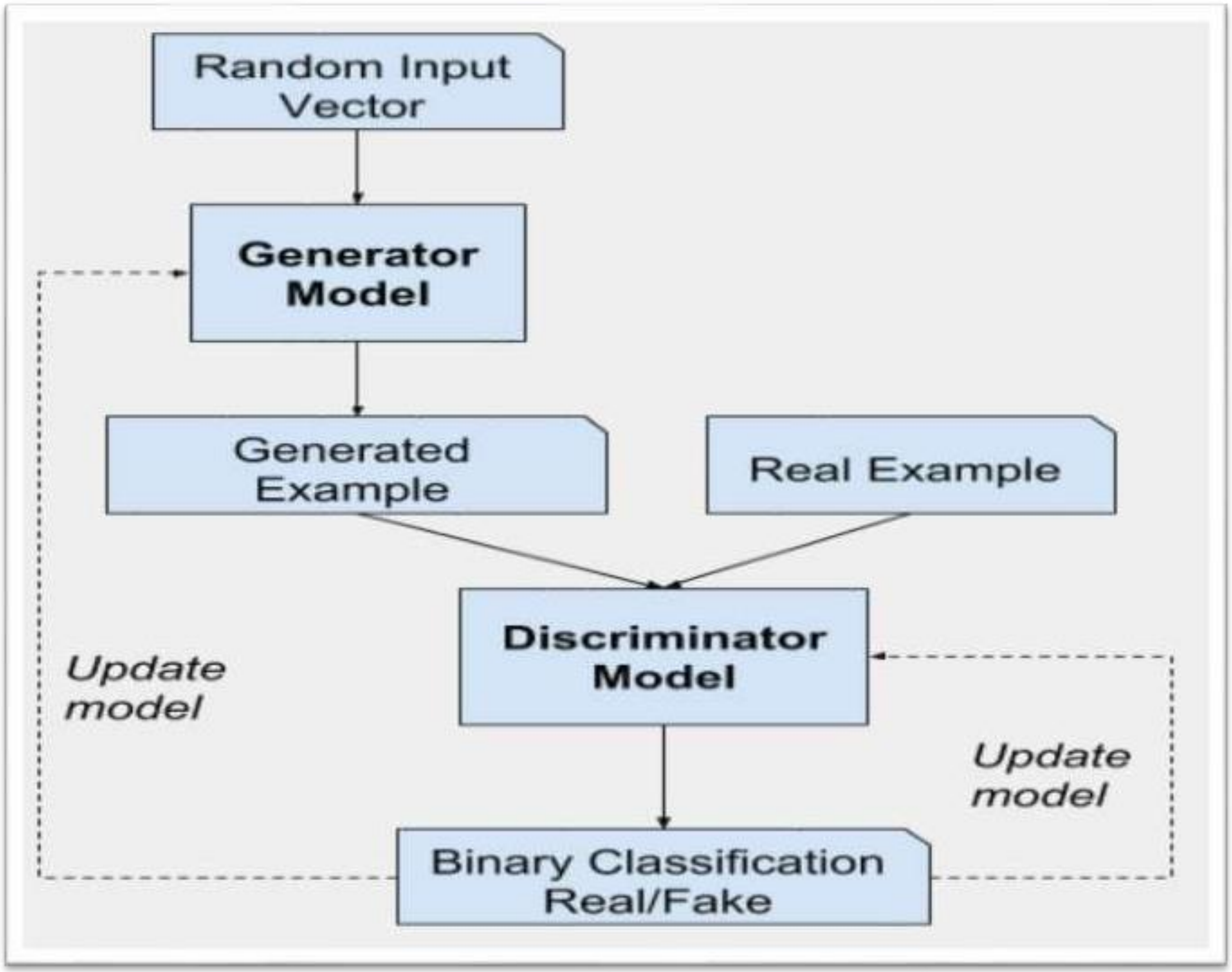


Figure 1: Backpropagation in generator training.

Flowchart Making



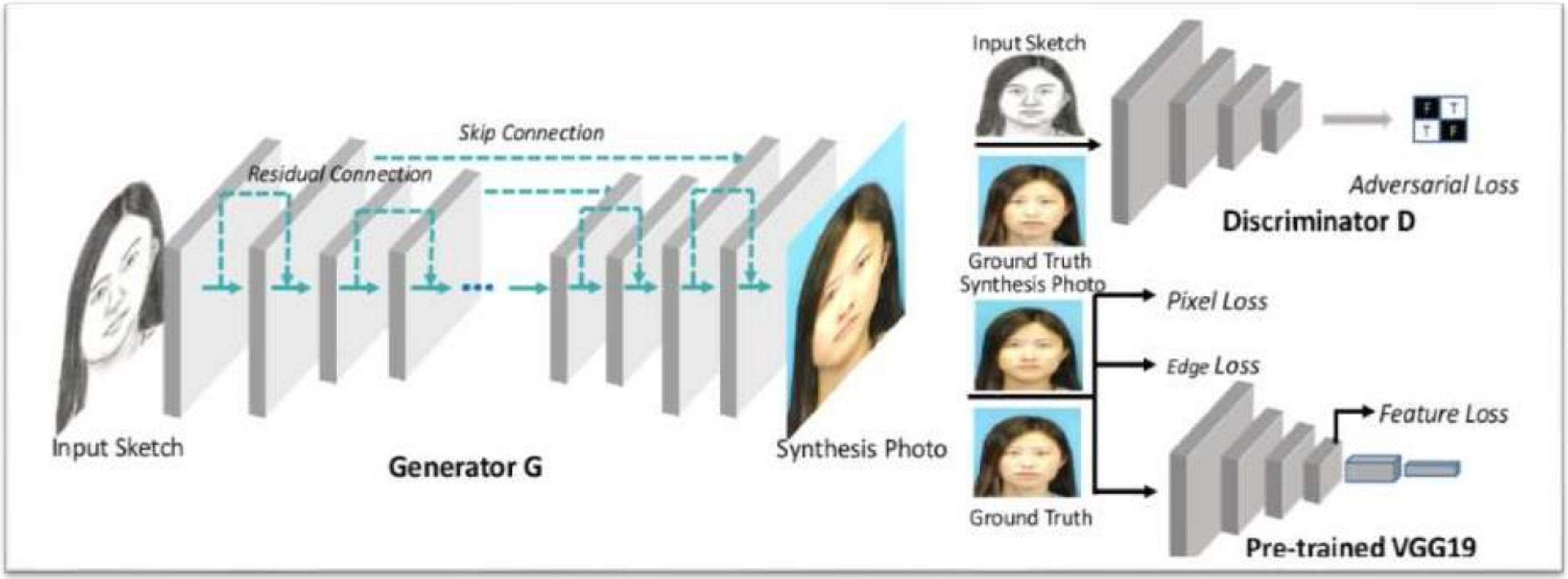
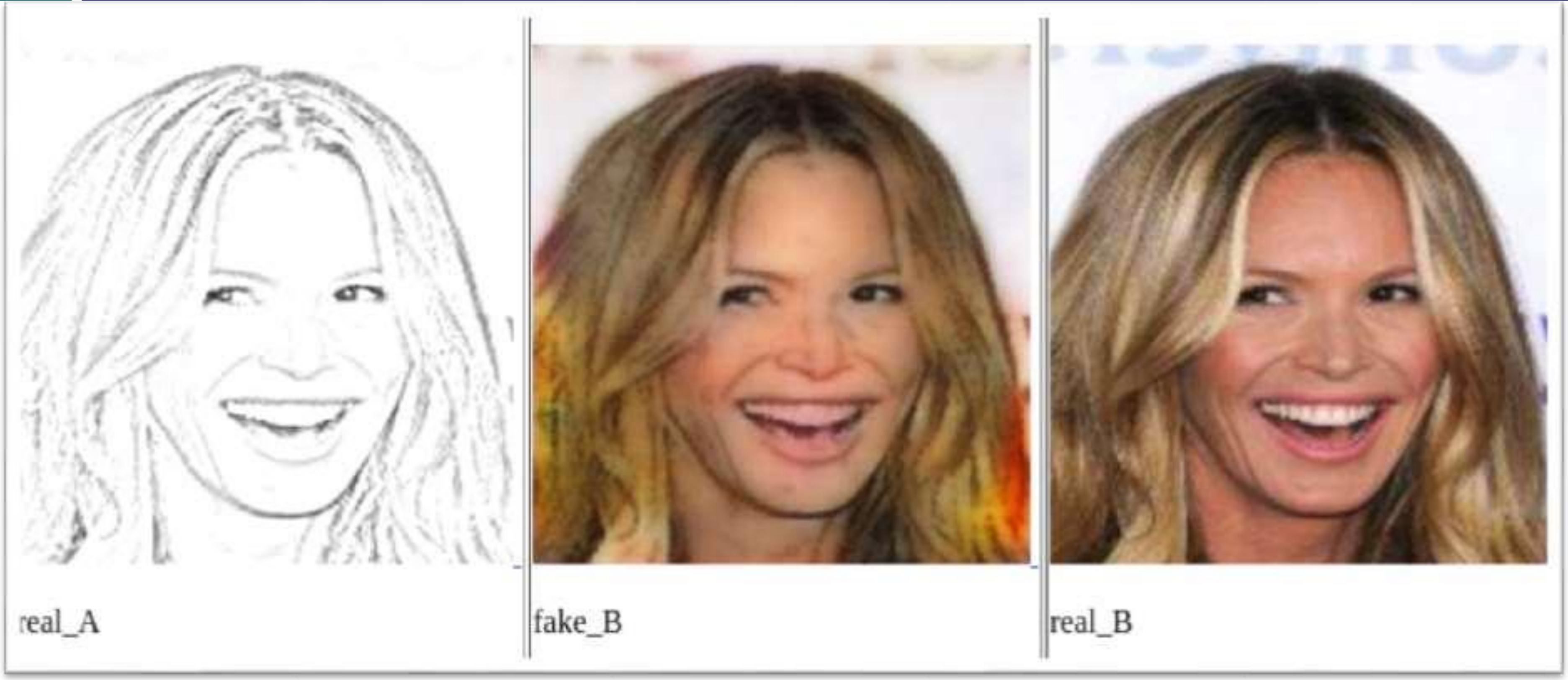


Figure: Police Sketch to Image by Gans's (Generative Adversarial Networks)

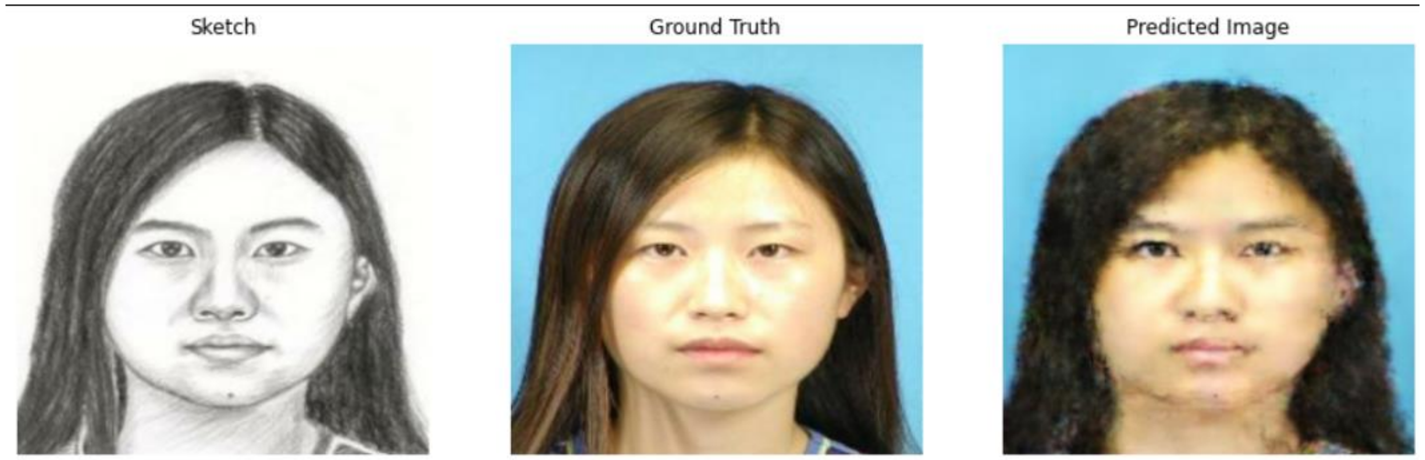

real_A
fake_B
real_B

Coding Explanation

1. **Data Preparation:** Preprocess sketches and images.
    - **Normalization**: Convert all images and sketches to a consistent size, usually 64x64 or 128x128 pixels.
    - **Normalization**: Scale pixel values to the range [-1, 1] for better GAN performance.
2. **GAN Architecture:** Design the Generator and Discriminator models.
    - **Generator**:- The Generator network takes a sketch as input and outputs a generated image.
    - **Discriminator**:- The Discriminator network takes an image (either real or generated) and outputs a probability indicating whether the image is real or fake.
3. **Training:** The GAN training involves training the Discriminator to distinguish real images from generated images and the Generator to produce images that the Discriminator cannot distinguish from real images.
4. **Generation:** Use the trained Generator to produce images from new sketches**.**

Implementation

On Jupyter Notebook, Vs code and Google Collab



Thank You