Face to Sketch Using GAN

Muhammad Abdur Rafey
Computer Science
FAST NUCES Islamabad
Islamabad, Pakistan
i21070@nu.edu.pk

***Abstract*—This paper presents the architecture and training process of a Conditional Generative Adversarial Network (cGAN) designed to generate realistic face images from input sketches. The Generator in this model processes the sketch through convolutional layers to capture key visual features and combines these with a latent noise vector to introduce variability in the output. The Generator then upsamples the combined feature representation to produce a 64x64 RGB image. The Discriminator serves as a binary classifier that distinguishes between real and generated images by processing both the sketch and the face image through convolutional layers. It combines the extracted features to determine whether the input image is real or generated. Both the Generator and Discriminator are trained adversarially, where the Generator strives to create indistinguishable images from real ones, and the Discriminator improves its classification accuracy. This architecture effectively leverages the synergy between the sketch input and noise for realistic face generation, while the adversarial training optimizes the overall performance of the model.**

***Index Terms*—Conditional GAN, Sketch, Realistic**

I. Introduction

This project focuses on translating human face sketches into realistic photographs using deep learning techniques. Specifically, the goal is to develop a Conditional Generative Adversarial Network (cGAN) that can generate accurate and lifelike face images based on corresponding sketch inputs. Traditional image-to-image translation tasks involve converting one type of image to another, but translating a simple sketch into a detailed photograph presents a unique challenge. The sketches typically lack the rich detail, color, and texture present in real photographs, making it difficult for a machine to accurately fill in the missing information.

II. Overview

The task involves training a Generator to create real-looking face images from sketch inputs while simultaneously training a Discriminator to distinguish between real photos and the images generated by the model. The adversarial nature of the two models helps improve the quality of the generated images over time. This problem has practical applications in fields such as art, design, and forensic science, where turning sketches into realistic representations can enhance the creative process or assist in investigations.

The challenge is to create a model that can learn to understand the relationship between sketches and real faces and generate high-quality, detailed images that align closely with the input sketches.

III. Generator Architecture

The Generator is responsible for generating realistic face images based on a sketch and random noise. Its architecture consists of the following key components:

A. Condition Processor (Sketch Input)

This part processes the input sketch through a series of convolutional layers. It reduces the spatial dimensions of the sketch while capturing important visual features. After passing through two convolutional layers with Leaky ReLU activation and batch normalization, the sketch is represented as a feature map of size $128 \times 16 \times 16$.

B. Noise Processor (Latent Space Input)

The Generator also takes in a random noise vector (latent space input), which provides variability to the generated images. This noise is passed through a fully connected layer, reshaped into a feature map, and prepared to be combined with the processed sketch.

C. Combining Condition and Noise

The processed sketch features and the noise vector are concatenated along the channel dimension, forming a combined feature map of size $256 \times 16 \times 16$.

D. Decoder

The combined feature map is passed through a series of transposed convolutional layers (deconvolution layers) to progressively upsample it to the original image size (64×64). At each stage, the model applies batch normalization and Leaky ReLU activation to maintain stability during training. The output layer uses a Tanh activation function, resulting in a 3-channel (RGB) image of size 64×64.

IV. Discriminator Architecture

The Discriminator acts as a binary classifier that tries to distinguish between real images and the ones generated by the Generator. It processes both the sketch and the image (either real or generated). The architecture of the Discriminator consists of:

A. Condition Processor (Sketch Input)

Similar to the Generator, the Discriminator processes the input sketch using convolutional layers. These layers reduce the spatial size of the sketch while capturing the key features, outputting a feature map of size $128 \times 16 \times 16$.

B. Image Processor (Real or Generated Image)

The real or generated face image (RGB) is passed through a separate set of convolutional layers. This step extracts key visual features from the image and outputs a feature map of the same size as the processed sketch ($128 \times 16 \times 16$).

C. Combining Condition and Image

The processed sketch features and the image features are concatenated along the channel dimension, creating a combined feature map of size $256 \times 16 \times 16$.

D. Feature Processing

The combined feature map is further processed through a convolutional layer that reduces its dimensions and extracts deeper features, resulting in a feature map of size $512 \times 8 \times 8$.

E. Classification

The feature map is flattened into a single vector and passed through a fully connected layer, followed by a Sigmoid activation function, which outputs a value between 0 and 1. This value represents the probability of the image being real.

V. Training Process (Adversarial Training)

The Generator and Discriminator are trained in an adversarial manner:

- The Generator tries to produce images that are indistinguishable from real photos.
- The Discriminator learns to differentiate between real and generated images.

During training, the Generator's goal is to fool the Discriminator by improving the quality of the generated images. The Discriminator, in turn, tries to correctly classify the real images as real and the generated ones as fake.

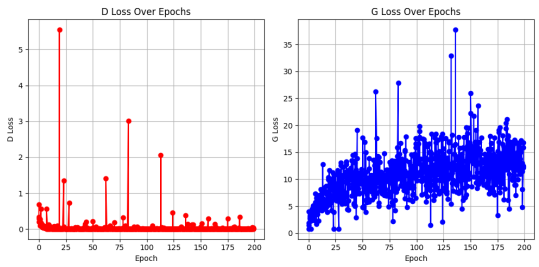

Fig. 1: Loss over epoch

VI. Dataset

The dataset should be organized into training and validation folders with paired face photos and corresponding sketches. You can use any face-sketch dataset or create your own by manually pairing photos and sketches. Make sure to name the photo and its corresponding sketch with the same file name (e.g., `image_01.jpg` in both `photos` and `sketches` directories).

Used dataset is from Kaggle, which can be found at: *https://www.kaggle.com/datasets/almightyj/person-face-sketches*.