Sketch-to-Image Translation using CycleGAN & Pix2Pix

Rushiraj Jadeja
Vandan Patel
Manjur Kovadiya

April 7, 2025

Problem Definition

Sketch-to-Image translation aims to convert input sketches into photorealistic images.

1. This is highly useful in the fashion industry where designers can visualize products from initial sketches.
2. This is also useful for converting sketches to colored images which helps in visualizing faces from sketch to real.

Objective

Our objective is to train a CycleGAN model to learn mapping between sketch images and their corresponding real images in a fashion dataset. The ultimate goal is to generate realistic outputs from raw sketches. Other model we trained is Pix2Pix which takes paired images and generates the real face images from the given sketch images.

Dataset Overview

For Fashion Model

- Contains paired/unpaired sketch and real images
- Separate ‘train/val‘ folders for both domains
- Images resized to 256x256

For Face Model

- Contains paired sketch and real images
- Has only one folder with the paired dataset
- Images resized to 256x256

CycleGAN: Brief Overview

- Two Generators: $G_{S \to R}$ and $G_{R \to S}$
- Two Discriminators: D_S and D_R
- Cycle-consistency loss ensures mapping is preserved
- Identity loss stabilizes learning

Generators and Discriminators

Generators

- $G_{S \to R}$: Converts **Sketch (S)** to **Real Image (R)**
  This is your main goal: turning fashion sketches into realistic clothes images.
- $G_{R \to S}$: Converts **Real Image (R)** back to **Sketch (S)**
  This is for **cycle-consistency**, ensuring that the generated image retains the structure of the original sketch.

Discriminators

- D_R: Distinguishes between **real** images and **fake/generated** images from $G_{S \to R}$
  Helps $G_{S \to R}$ generate more **realistic** outputs.
- D_S: Distinguishes between **real** sketches and **fake** sketches from $G_{R \to S}$
  Helps $G_{R \to S}$ learn reverse mapping and support the cycle.

Training Details

- Losses: GAN + Cycle Consistency + Identity
- Optimizer: Adam (lr=0.0001, β_1=0.5, β_2=0.999)
- **Beta1 (0.5)**: Controls the exponential decay rate for the **first moment estimates** (mean of gradients).
  $\Rightarrow$ Lower than the default value of 0.9, this helps **stabilize GAN training** by reducing oscillations.

- **Beta2 (0.999)**: Controls the exponential decay rate for the **second moment estimates** (variance of gradients).
  $\Rightarrow$ Kept close to 1, as it is the **standard setting** and performs well in most deep learning tasks.
- Early stopping used for stability
- Output saved every 5 epochs

Code Structure: Setup Model

- Define Generator and Discriminator
- Initialize weights
- Use CycleGAN model class
- Define losses: MSE + L1

Code Structure: Training Loop

- Forward pass through generators
- Calculate cycle and identity loss
- Backpropagate generator losses

Code Structure: Discriminator Training

- Train D_S and D_R on real and generated images
- Use replay buffer to stabilize training
- Use adversarial loss for backpropagation

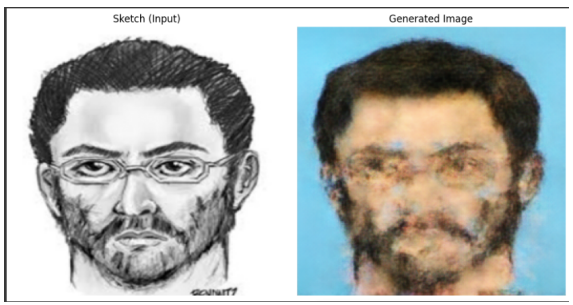
Code Structure: Saving Evaluation

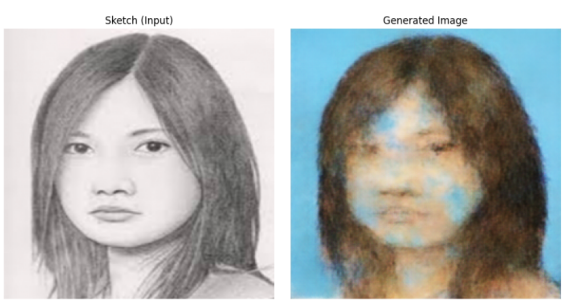
- Save model checkpoints
- Save generated samples every 5 epochs
- Plot loss curves
- Apply early stopping if no improvement

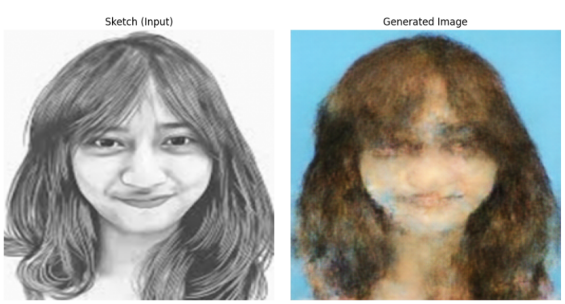
Performance Evaluation

- Visually assess quality of generated images
- Training loss decreases steadily
- Generated images become more realistic with epochs

Output


Sketch (Input)
Generated Image


Sketch (Input)
Generated Image


Sketch (Input)
Generated Image

Conclusion

- Successfully translated sketches to realistic images
- CycleGAN and Pix2Pix Models learned mappings from limited data
- Can be improved using better architectures and FID score evaluation

Q&A

Thank you! Questions?