Sketch-to-Photo Generation Using GANs

A deep learning pipeline to convert face sketches into realistic photo images using PyTorch

Overview.

This project focuses on transforming human face sketches into full-color photo-realistic images using a Generative Adversarial Network (GAN). The generator learns to map sketches to RGB photos while the discriminator helps refine realism. Built entirely in PyTorch, this model trains on paired datasets of sketch-photo images.

Objective.

- Train a GAN to reconstruct photo images from sketch input.
- Use L1 loss and adversarial feedback for better image fidelity.
- Generalize across validation images using normalized training data.

Tools.

- **PyTorch** (deep learning engine)
- **Torchvision** (transformations, dataloaders)
- **Matplotlib** (visual outputs)
- **PIL** (image loading)
- **CUDA (GPU)** acceleration
- **Custom PyTorch Dataset class**

Model architecture.

Generator
- Encoder-decoder CNN
- 3 Conv layers → 3 Deconv layers
- Uses BatchNorm, ReLU, and Tanh
- Converts grayscale sketches to RGB outputs

Discriminator
- CNN with LeakyReLU activations
- Classifies real vs generated images
- Input: sketch + photo combined as 4 channels

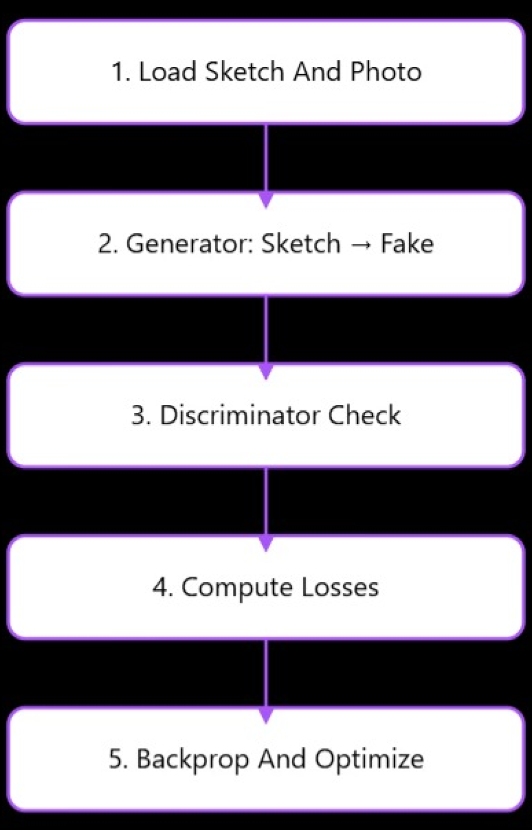
Training Strategy.

- **Adversarial Loss:** BCELoss to trick the discriminator
- **Pixel Loss:** L1Loss to improve image similarity
- **Optimization:** Adam (β_1=0.5, β_2=0.999)
- **Epochs:** 100
- **Display:** Side-by-side sketch vs generated image per epoch

Challenges & Process

The project began with constructing a custom PyTorch dataset loader to efficiently pair grayscale sketches with their corresponding RGB photos. The training pipeline leveraged an encoder-decoder style generator and a patch-based discriminator to enable end-to-end sketch-to-photo synthesis. Images were normalized and transformed before training. One of the key challenges was ensuring that fine details—such as facial contours, eyes, and hairlines—remained consistent during generation. Early training stages produced blurry outputs, requiring careful tuning of the learning rate, loss weighting, and model architecture to avoid mode collapse. The discriminator often overpowered the generator, necessitating balanced adversarial training. Finally, real-time visualization after each epoch played a critical role in tracking convergence and correcting instability during training.

Flowchart



Results, Learnings, and Impact

The model successfully generated realistic face photos from sketches, capturing fine facial structure, contours, and lighting consistency. The adversarial loss helped sharpen edges and textures, while L1 loss ensured the output stayed faithful to the input sketch. Key learnings include the effectiveness of combining pixel-based and adversarial objectives, the importance of data normalization, and how custom PyTorch dataset loaders enable precise control over input-output alignment. Additionally, the project deepened understanding of low-level tensor operations, conditional GAN training strategies, and image transformation pipelines.