Sketch-to-Image Translation Using GANs

# 1. Objective

The aim of this project is to develop a deep learning model that can translate input sketches into corresponding realistic images. This is particularly useful in applications like product design, fashion, and human-computer interaction, where sketches need to be converted into visual prototypes for better interpretation and analysis.

# 2. Dataset Description

We used the UT Zappos50K dataset, which contains over 50,000 shoe images. Although the dataset does not directly provide sketches, it was adapted by treating grayscale or edge-detected versions of the images as sketches and using the original colored images as target outputs.

The dataset was downloaded from the official UT Zappos50K Project Website and extracted using standard command-line utilities.

# 3. Methodology

A Conditional Generative Adversarial Network (cGAN) was employed to handle the image translation task. The overall architecture consists of:

- Generator: A U-Net-based generator that converts input sketches into realistic images. The skip connections in U-Net architecture help retain spatial information, crucial for preserving outlines and structure.  
   
- Discriminator: A PatchGAN-based discriminator that judges the realism of the generated image conditioned on the input sketch. Patch-based classification ensures better attention to local details.

# 4. Loss Functions

To ensure both pixel-level accuracy and perceptual quality, the following losses were used:

- Adversarial Loss (BCEWithLogitsLoss): Encourages the generator to produce realistic outputs that can fool the discriminator.

- Pixel-wise L1 Loss: Penalizes differences between the generated image and the real target image at the pixel level.

- Perceptual Loss (VGG-based): Uses features extracted from a pretrained VGG19 network to compare generated and real images at a higher semantic level, improving the visual fidelity.

# 5. Training Details

- The model was implemented in PyTorch.  
- A custom Dataset class was written to handle image preprocessing (resizing, grayscale conversion for sketches, normalization).  
- Separate optimizers were used for the generator and discriminator.  
- The training process utilized GPU acceleration (`torch.cuda`) for efficient computation.  
- Periodic saving of generated outputs was used to monitor training progress.

- The training was done for 20 epochs.

Fig 1: The Training Process

# 6. Results

Below are sample outputs from the trained model showing:

- Input Sketch  
- Generated Image  
- Ground Truth Image

These comparisons highlight the model’s ability to learn and reproduce key details and colors from the sketch input.

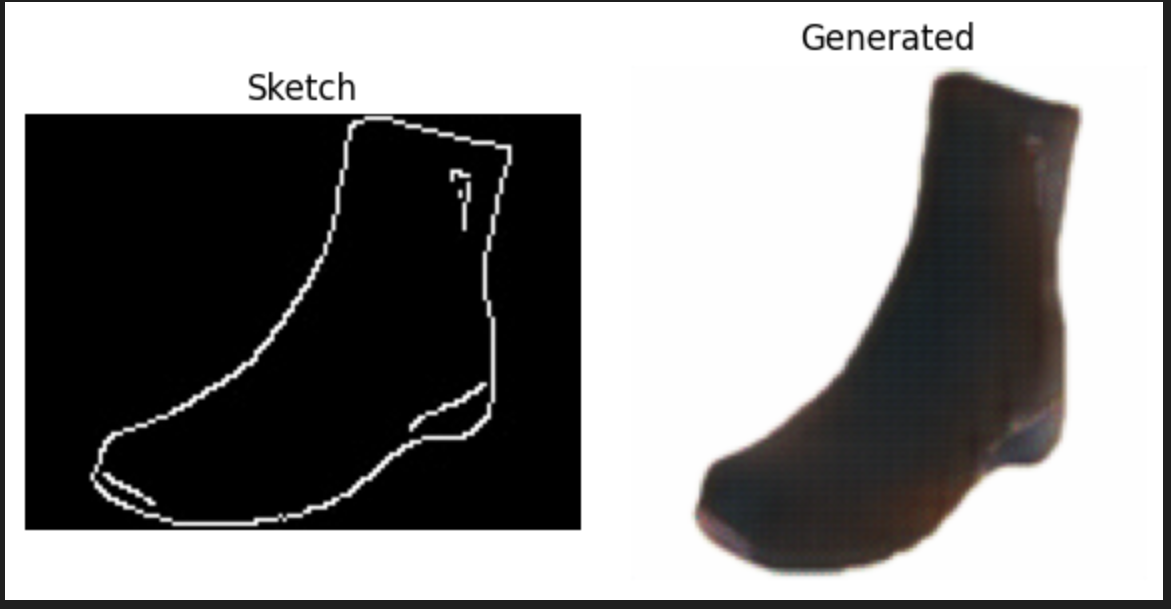


Fig 2: The Output

The outputs show promising visual quality with strong alignment to ground truth shapes and color patterns. However, finer details and texture realism can be further improved with extended training and architectural tweaks.

# 7. Conclusion

This project successfully demonstrates the use of a cGAN-based architecture for sketch-to-image translation using the UT Zappos50K dataset. The use of perceptual loss in combination with adversarial and pixel losses helped improve both semantic and visual accuracy of the generated images.

# 8. Future Work

- Improve sketch generation using edge detection techniques like Canny or HED for better alignment with real-world sketches.  
- Extend the model to work with multi-category datasets (e.g., clothing, accessories).  
- Experiment with attention modules or vision transformers to capture long-range dependencies in images.  
- Integrate user-controlled color hints or interactive generation features.