Virtual Try-On Model Training Workflow using Kaggle Dataset

Overview

This document provides a detailed workflow to train a Virtual Try-On (VTON) model using the Kaggle dataset: https://www.kaggle.com/datasets/adarshsingh0903/virtual-tryon-dataset. The structure is modeled after pipelines like CP-VTON, VITON, and others used in virtual garment try-on tasks.

1. Dataset Structure

```
virtual_tryon_dataset/
├── train/
│   ├── cloth/
│   ├── cloth-mask/
│   ├── image/
│   ├── openpose_img/
│   └── openpose_json/
├── test/
│   ├── cloth/
│   ├── cloth-mask/
│   ├── image/
│   ├── openpose_img/
│   └── openpose_json/
├── train_pairs.txt
└── test_pairs.txt
```

- **cloth/**: Garment images (standalone)
- **cloth-mask/**: Binary mask of garments
- **image/**: Person images wearing original garments
- **openpose_json/**: JSON keypoints from OpenPose
- **openpose_img/**: Visual representation of pose (optional)
- **train/test_pairs.txt**: Text files containing image pair mappings

2. Prerequisites and Setup

Tools Required:

- Python >= 3.7
- PyTorch
- OpenCV
- NumPy

Installation

```
git clone https://github.com/sergeywong/cp-vton.git
cd cp-vton
pip install -r requirements.txt
```

3. Data Preprocessing

Resize Images

Resize all images to 256x192:

```
import cv2
import os

for folder in ['image', 'cloth', 'cloth-mask']:
    path = f"train/{folder}"
    for file in os.listdir(path):
        img = cv2.imread(os.path.join(path, file))
        img = cv2.resize(img, (192, 256))
        cv2.imwrite(os.path.join(path, file), img)
```

Pose Map Generation (Optional)

Extract pose keypoints from OpenPose JSON and convert to heatmaps.

4. DataLoader

Create a custom `Dataset` class in PyTorch to:

- Read image pairs from `train_pairs.txt`
- Load corresponding person image, cloth, mask, pose map

5. Model Pipeline

Stage 1: Geometric Matching Module (GMM)

- **Input**: Person Image + Pose + Cloth
- **Output**: Warped Cloth Image
- **Loss**: L1 loss

Train GMM:

```
python train_gmm.py --dataset train --batch_size 8 --lr 0.0001
```

Stage 2: Try-On Module (TOM)

- **Input**: Person Image + Warped Cloth + Mask
- **Output**: Final Synthesized Image
- **Loss**: Perceptual + L1 loss

Train TOM:

```
python train_tom.py --dataset train --batch_size 8 --lr 0.0001
```

6. Inference

Generate try-on images:

```
python test.py \
    --dataset test \
    --checkpoint_gmm checkpoint/gmm_final.pth \
    --checkpoint_tom checkpoint/tom_final.pth
```

7. Output

Each output will be a synthesized image of a person wearing the new garment defined in the test pairs.

8. Tips for Success

- Ensure perfect alignment between pairs.txt and actual files
- Normalize input images between [-1, 1]
- Use data augmentation for generalization

9. Optional Enhancements

- Switch to VITON-HD or Diff-VTON for higher realism
- Use segmentation maps for better person-cloth fusion
- Use LPIPS and SSIM metrics for evaluation

Summary

This workflow enables end-to-end training and inference of a virtual try-on model using the provided Kaggle dataset. For advanced performance, consider experimenting with GAN-based fusion networks or diffusion-based synthesis.