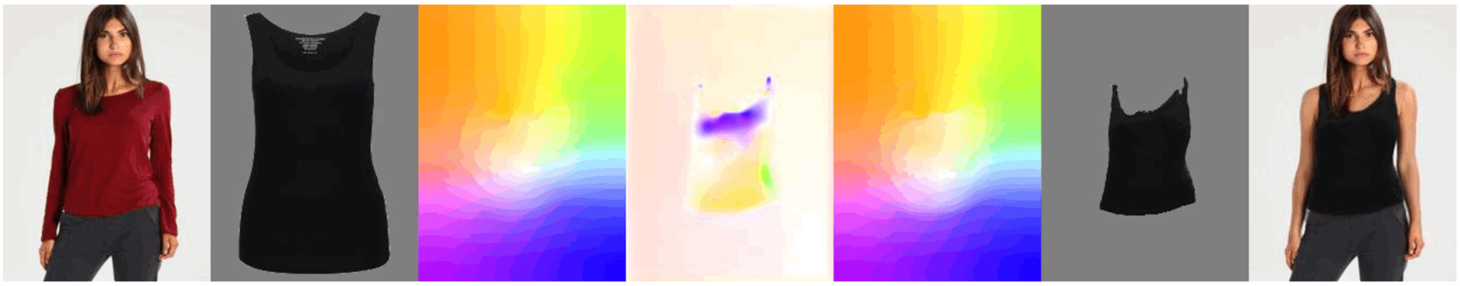
Style-Based Global Appearance Flow for Virtual Try-On (CVPR 2022)

Project Page | Paper | Video | Poster | Supplementary Material



Requirements

- python 3.6.13
- torch 1.1.0 (as no third party libraries are required in this codebase, other versions should work, not yet tested)
- torchvision 0.3.0
- tensorboardX
- opencv

Inference (`cd` to test folder)

Download the testing data from here.

If you want to test on the augmented testing images, download the images from here and put it to the testing data folder downloaded above, change the image folder here to `_ma_img` .

Download pretrained checkpoints from here.
The checkpoint trained without augmentation is better for testing set in the VITON. But the checkpoint trained with augmentation is more robust for in-the-wild images.

```
python test.py --name demo --resize_or_crop None --batchSize 1 --gpu_ids 0 --warp_checkpoint your_path_to_the_dow
```

Test

FID and SSIM

Training (`cd` to the train folder)

For VITON dataset, download the training data from VITON_train and put the folder `VITON_traindata` under the folder `train/dataset`

For perceptual loss computation, download the vgg checkpoint from VGG_Model and put `vgg19-dcbb9e9d.pth` under the folder `train/models` .

Custom dataset

For other custom dataset, please generate the training data folder with the same structure as `VITON_traindata` .

More specifically, you need to prepare human parsing, pose (18 key points, saved in .json file) and densepose (a heatmap, different body region has different value).

For human parsing, you can use Human parser.

For pose, you can use Openpose.

For dense pose, you can use Dense pose.

Note: To train the model on your own dataset, you may need to decrease the learning rate to 0.00001.

Note: if you want to train with augmentation, check here, here, here and here in the code.

Stage 1: Parser-Based Appearance Flow Style

```
sh scripts/train_PBAFN_stage1_fs.sh
```

Stage 2: Parser-Based Generator

```
sh scripts/train_PBAFN_e2e_fs.sh
```

Stage 3: Parser-Free Appearance Flow Style

```
sh scripts/train_PFAFN_stage1_fs.sh
```

Stage 4: Parser-Free Generator

```
sh scripts/train_PFAFN_e2e_fs.sh
```

Reference

If you find this repo helpful, please consider citing:

```
@inproceedings{he2022fs_vton,
  title={Style-Based Global Appearance Flow for Virtual Try-On},
  author={He, Sen and Song, Yi-Zhe and Xiang, Tao},
  booktitle={CVPR},
  year={2022}
}
```

Acknowledgements

This repository is based on PF-AFN, where we replaced the tensor correlation based flow estimation with our proposed style-based flow estimation.