Deep Learning for Virtual Tryon Clothes

Overview:

The goal of the project was to build a Virtual Tryon application. Here the aim is to assist individuals to make right purchase decisions by enabling them to virtually check out clothing on factors such as size, fit and style. It involves mapping a virtual realistic representation of the clothing over an avatar to depict how the product would look. Also, there is the factor of ease in changing clothes and colors while exploring options.

The specific need of our client was an application that would enable virtually checking out of tops on an avatar i.e. while the avatar would remain fixed, the clothing could be changed at will. They wanted to sell this application to their customers who were mostly big ecommerce players involved in apparel merchandising.

The application is such that in a matter of seconds, tentative buyers are offered realistic and personalized presentations to assist their buying decision. For the sellers, the benefit is twofold. Besides increasing the possibility of selling, they save significantly on expenses; by diminishing the cases of product-returns. The project had 2 primary goals. First was to build a deep neural network that could swap a high resolution input garment onto a high resolution input model. Second was to speed up the training process which otherwise generally takes a few months to reach a reliable and stable state.

Client details:

Name: Confidential **|** **Industry**: Publishing **|** **Location**: USA

Technologies:

Deep Learning, Image Segmentation, Image Processing, Pose Detection (OpenPose), PyTorch (Python) OpenCV, Pandas, NumPy, Matplotlib, Docker, CUDA

Deep Learning for Virtual Tryon Clothes

Project Description:

An Algorithm was used that leveraged Generative and Adversarial Networks (GANs) to train the model over a large dataset (basically consisting of on-figure and laydown pairs).

A Generative Adversarial Network (GAN) is a deep learning architecture that consists of two neural networks competing against each other in a zero-sum game framework. GANs are basically made up of a system of two competing neural network models which compete with each other and are able to analyze, capture and copy the variations within a dataset. They are highly effective in image synthesis and manipulation. The two competing neural networks are a generator and a discriminator. The generator learns to generate realistic images to deceive the discriminator, while the discriminator learns to distinguish the synthesized images from the real ones. This interplay can be actively put to effective use on tasks such as style transfer, image in painting, and image editing. The extensive applications of GAN further demonstrate its superiority in image synthesis.

The algorithm for this project uses GANs to train the model over a large dataset (basically consisting of on-figure and laydown pairs). We used an existing GAN model that consists of 3 GAN networks and 1 UNET and made changes on top of it, making it adaptive to our requirements and achieving our end goal.

Model Architecture

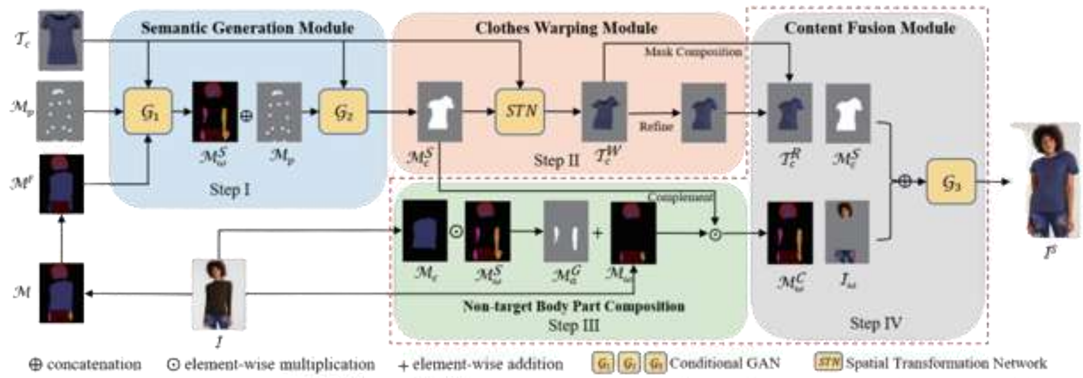
The model basically consists of 3 GAN modules (G1, G2, G3), each built upon networks.
First module is the Semantic Generator - G1 network
Second module is the Cloth Warping Module - G2 network + STN optimization
Third module is the Fusion Module - G3 network
The training module keeps G1 and underlying UNET fixed after 20 epochs and the rest 2 are trained for longer.

The preparation process involves broadly two steps:

1. Data Preprocessing
2. Training

Deep Learning for Virtual Tryon Clothes

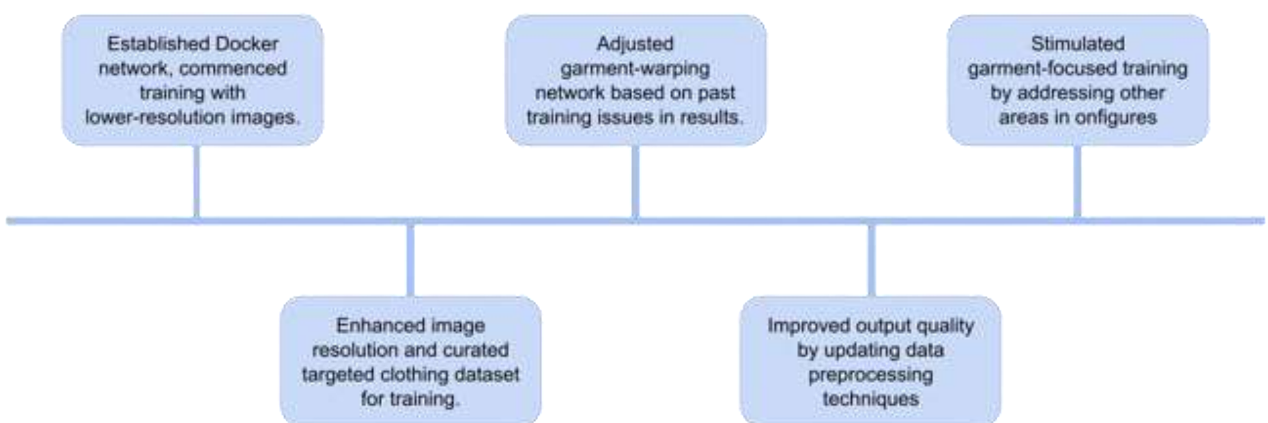
Training Pipeline



The overall architecture of our ACGPN. (1) In Step I, the Semantic Generation Module (SGM) takes target clothing image $\mathcal{T}_c$, the pose map $\mathcal{M}_p$, and the fused body part mask $\mathcal{M}^F$ as the input to predict the semantic layout and to output synthesized body part mask $\mathcal{M}_\omega^S$ and target clothing mask $\mathcal{M}_c^S$; (2) In Step II, the Clothes Warping Module (CWM) warps the target clothing image to $\mathcal{T}_c^R$ according to the predicted semantic layout, where a second-order difference constraint is introduced to stabilize the warping process; (3) In Steps III and IV, the Content Fusion Module (CFM) first produces the composited body part mask $\mathcal{M}_\omega^C$ using the original clothing mask $\mathcal{M}_c$, the synthesized clothing mask $\mathcal{M}_c^S$, the body part mask $\mathcal{M}_\omega$, and the synthesized body part mask $\mathcal{M}_\omega^S$ and then exploits a fusion network to generate the try-on images $\mathcal{I}^S$ by utilizing the information $\mathcal{T}_c^R$, $\mathcal{M}_c^S$, and body part image $\mathcal{I}_\omega$ from previous steps.

Reference: https://arxiv.org/pdf/2003.05863.pdf

Training Roadmap



Deep Learning for Virtual Tryon Clothes

Variables impacting performance of the output

- Resolution of the Dataset
- Size of the dataset
- Diversity in the dataset
- Noise in the dataset
- Size of each GPU
- Batch size of training model
- GPU CUDA cores

Key challenges in optimization

- Speed up the training time from months to at least weeks
- Optimizing the training code to fix convergence issues as we are training using smaller batch size in the current environment
- We observe time taken for optimization step : 0.003-0.005 ms
- Time taken for backpropagation: 500-600 ms
- As we are using multiple GANs for the training, it becomes difficult to fit the whole model in one GPU and as a result of splitting the model over different GPUs, time and performance are affected