virtual try-on

High-Resolution Virtual Try-On via Misalignment-Aware Normalization

Reza Nematollahi AmirHossein Ezzati

abstract

The goal of image-based virtual try-on is to overlay a chosen clothing item onto the appropriate area of a person. This process typically involves adjusting the item to fit the target body part and seamlessly blending it with the person's image. Despite numerous studies in this field, the generated images remain at a low resolution (e.g., 256×192), which significantly hinders the ability to meet the expectations of online shoppers.

VITON-HD introduces an innovative virtual try-on technique that generates high-resolution images at 1024×768 pixels. The process begins with creating a segmentation map to guide the synthesis. Then, the target clothing item is roughly adjusted to fit the person's body. To address misalignments and maintain the details of the high-resolution inputs, the method incorporates ALIgnmentAware Segment (ALIAS) normalization and an ALIAS generator. Extensive comparisons with existing methods show that VITON-HD significantly outperforms them in both qualitative and quantitative measures of image quality.

In this project, we worked with VTON-HD model designed for virtual try-on applications. Our primary focus was on refining the pre-processing pipeline. This involved optimizing pose estimation, clothing segmentation and training human parsing model. We focused on making the process better and faster by trying out different ideas and testing quicker models. Our goal was to improve how we prepare the data before running the main part of the project.

Contents

Introduction 4

Methods: 4

Methods overview: 4

Pre-processing : 5

Pose-estimation: 5

Open pose: 6

Media pipe: 7

Clothes segmentation 7

Image parsing 7

Dataset: 8

Training: 9

Validation: 11

Performance: 12

Clothing-Agnostic Person Representation: 13

Segmentation Generation: 14

Clothing Image Deformation 14

Try-On Synthesis via ALIAS Normalization 16

Introduction

This paper maintains four main stages (pre-processing, Segmentation generator, Clothes deformation and Try-on synthesis) which we focused on pre-processing phase to optimize the speed and performance of this phase.

Pre-processing includes three steps

1 Pose-estimation
2 Image parsing
3 Cloth segmentation

In each step we used the best methods to achieve our goal.

Methods:

Methods overview:

Given a reference image ($\mathrm{I} \in \mathrm{R}^{3 \times \mathrm{H} \times \mathrm{W}}$) of a person and a clothing image ($\mathrm{c} \in \mathrm{R}^{3 \times \mathrm{H} \times \mathrm{W}}$) (where H and W denote the image height and width, respectively), the objective of VITON-HD is to generate a synthetic image ($\hat{I} \in R^{3 \times H \times W}$) of the same person wearing the target clothes (c), while preserving the pose and body shape from (I) and the details of (c). Although training the model with ($(I, c, \hat{I})$)triplets is straightforward, constructing such a dataset is costly. Instead, we use ((I, c, I)), where the person in the reference image (I) is already wearing (c).

Directly training on((I, c, I)) can negatively impact the model's generalization ability at test time. Therefore, we first create a clothing-agnostic person representation that excludes the clothing information in (I) and use it as input. This new clothing-agnostic person representation employs both the pose map and the segmentation map of the person to remove the clothing information from (I). The model generates the segmentation map from this clothing-agnostic person representation to aid in generating ($\hat{I}$).
Next, we deform (c) to approximately align it with the human body. Finally, we introduce the ALIgnment-Aware Segment (ALIAS) normalization to eliminate misleading information in the misaligned area after deforming (c). The ALIAS generator fills the misaligned area with the clothing texture and retains the clothing details.

Pre-processing :

Pose-estimation:

Pose estimation is a computer vision technique that determines the position and orientation of an object or person in an image or video. It involves identifying and tracking key points, such as joints on a human body or specific features on an object, to understand its spatial configuration.

1. Body25 (OpenPose):

Body25 is a pose estimation model provided by OpenPose, a popular open-source library for real-time multi-person detection.

- **Key Points**: 25 key points.
- **Details**: Includes 25 points covering the whole body, including facial key points, hands, and feet.
  Highly detailed and provides comprehensive coverage of human anatomy.
- **Use Cases**: Ideal for applications requiring detailed body movements, such as sports analysis, dance, and complex gesture recognition.

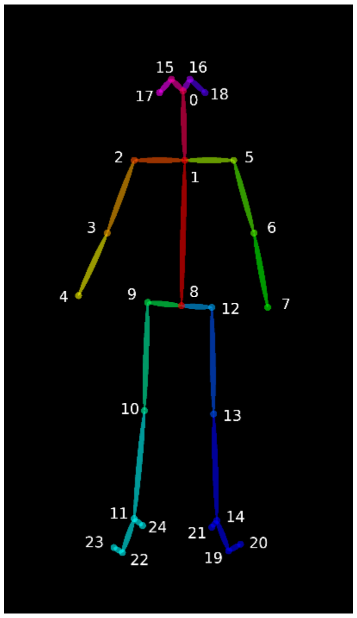

Figure 1, Body-25 format

2. COCO (Common Objects in Context):

COCO is a large-scale object detection, segmentation, and captioning dataset, which also includes a pose estimation model.

- **Key Points**: 17 key points.
- **Details**: Focuses on major joints such as the nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles.

Widely used in research and applications due to its balance between simplicity and effectiveness.

- **Use Cases**: General-purpose applications like human-computer interaction, video surveillance, and mobile applications.

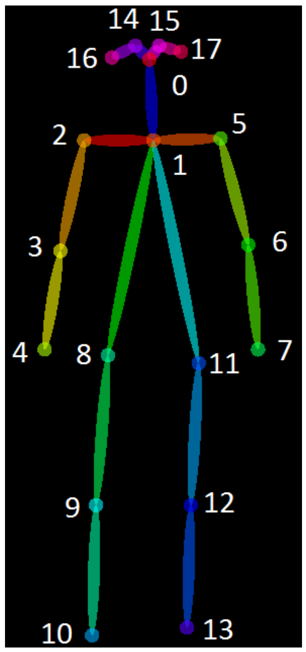

Figure 2, COCO format

3. MPII (Max Planck Institute for Informatics):

MPII Human Pose dataset is another widely used dataset for evaluating human pose estimation methods.

- **Key Points**: 16 key points.
- **Details**: Similar to COCO but with slightly different key point definitions and annotations.
  Focuses on body parts such as the head, neck, shoulders, elbows, wrists, hips, knees, and ankles.
- **Use Cases**: Suitable for activities where limb movements are crucial, such as action recognition and behavior analysis.

Between those three types of pose-estimations VTON-HD works with body25 format to achieve to this format we could use openpose project but due to several drawbacks of this project rather than mediapipe we decided to use mediapipe

Open pose:

- Written in C++ and needs to compile on computer.
- Doesn't use GPU to accelerate the process.
- 70 seconds per image

Media pipe:

- Written in C++ but you can use it with python.
- Use TensorFlow Lite.
- About 1 second per image

Clothes segmentation

Clothes segmentation is a computer vision task that involves partitioning an image into segments, specifically identifying and isolating different articles of clothing within the I mage. This process is typically achieved using machine learning techniques, particularly convolutional neural networks (CNNs) and more advanced deep learning architectures like U-Net, Mask R-CNN, or Fully Convolutional Networks (FCNs).
Due to various projects in this field we decided to use “cloths_segmentation” project to use which is fast and accurate enough.
This model takes about 1 second to segment an image.

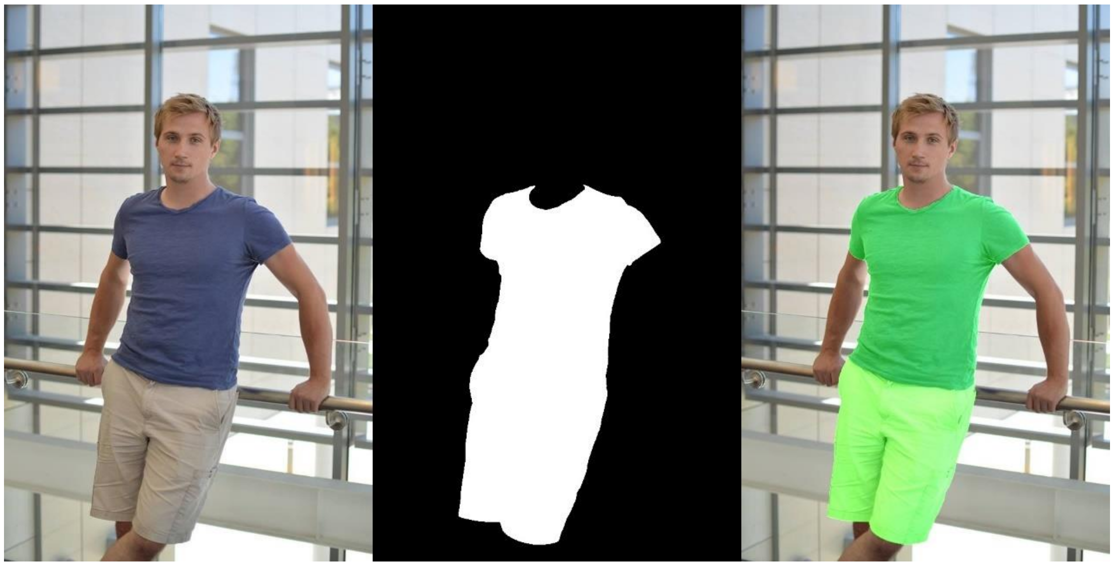

Figure 3, cloths_segmentation demo

Image parsing

Image parsing, also known as image segmentation or scene parsing, is a computer vision task that involves dividing an image into its constituent parts and identifying what each part represents. This process aims to assign a class label to each pixel in the image, effectively understanding the scene at a detailed level.

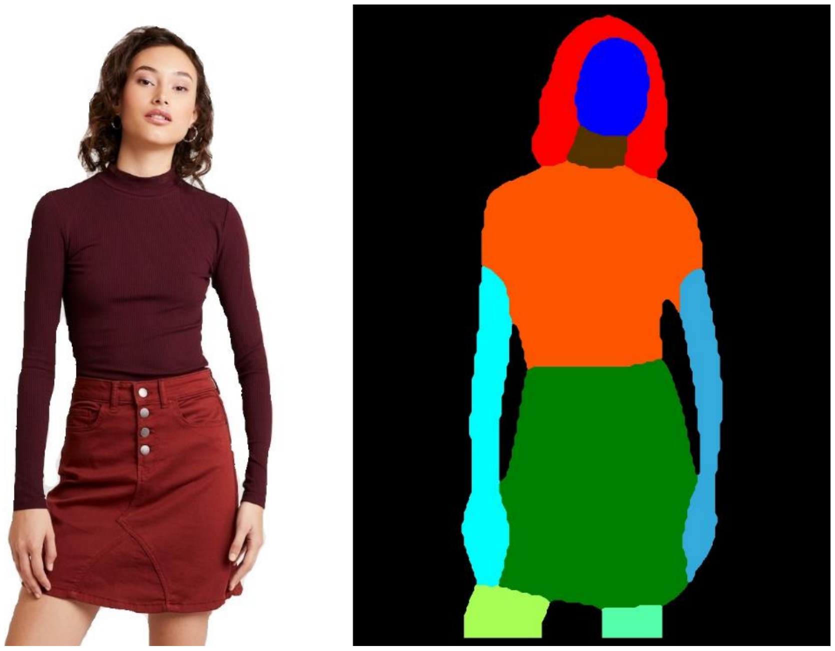

Figure 4, Human Image Parsing

There are too many projects on the internet for this purpose.
at first, we decided to try one of the popular projects LIP (Look into Person), but the output of was not good as we were expected.
so, we decided to train our model so we used YOLO V8 instance segmentation. We worked with YOLO V8x architecture.

Dataset:

Our dataset includes 1000 labeled images for training and 200 images for validation.
Each labeled image consists of set of body parts such as: hear, face, neck, upper clothes, right hand, left hand, trousers and skirt.

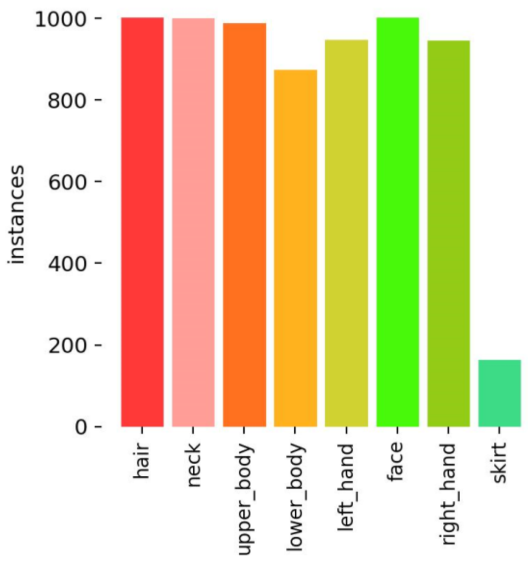

Figure 5, Number of labels in training dataset

Training:

We used Google Colab platform to train our model and it takes more than 10 hours on NVIDIA T4 Tensor Core GPU (for AI Inference) with 15 GB GPU RAM.

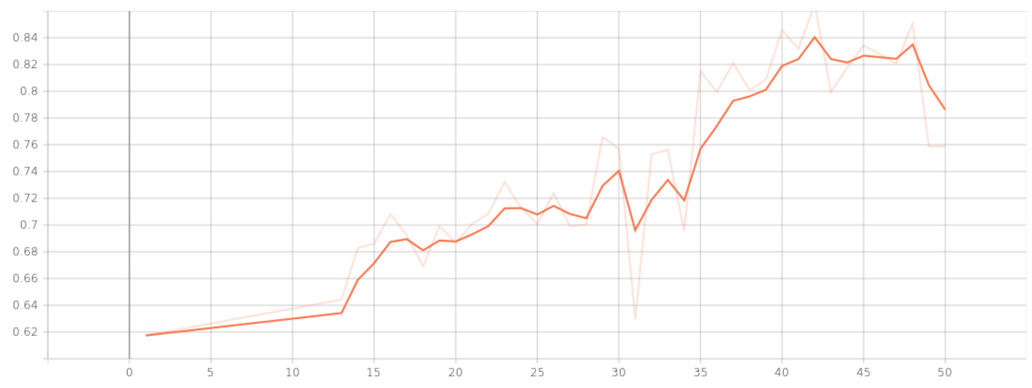

Figure 6 , precision (Box)

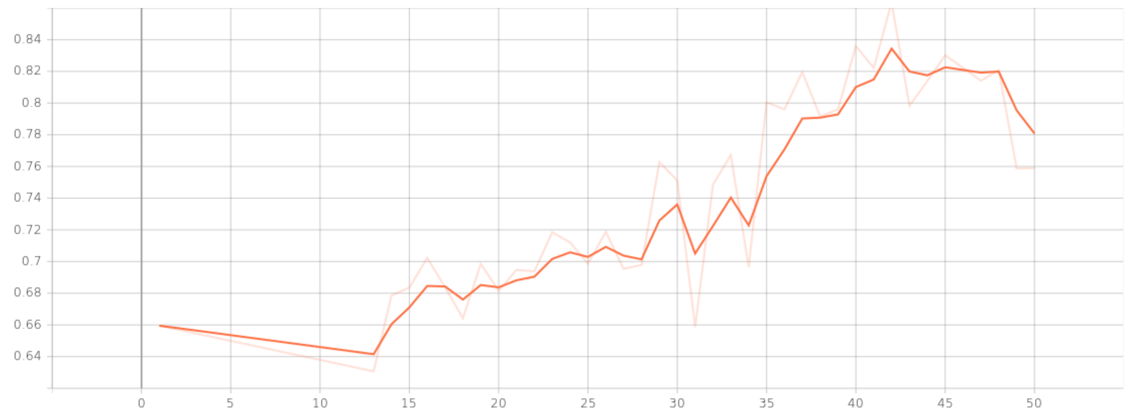

Figure 7, precision (Mask)

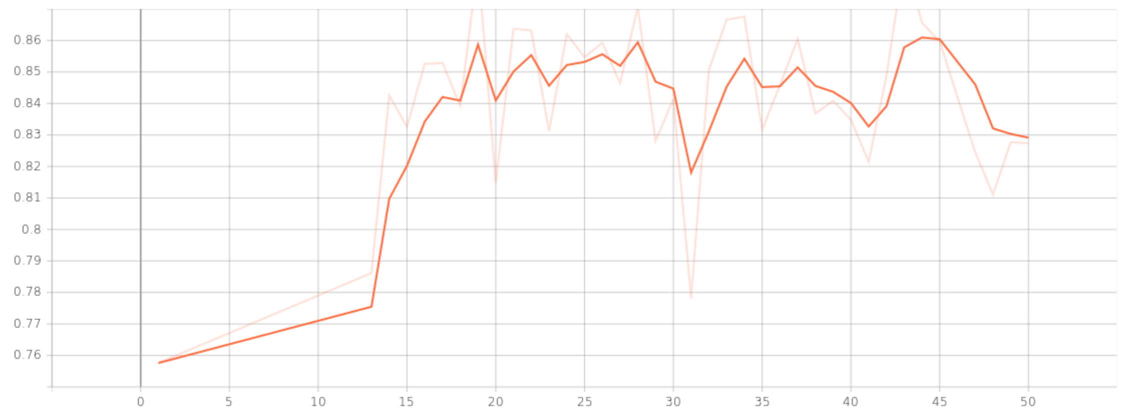

Figure 8, recall (Box)

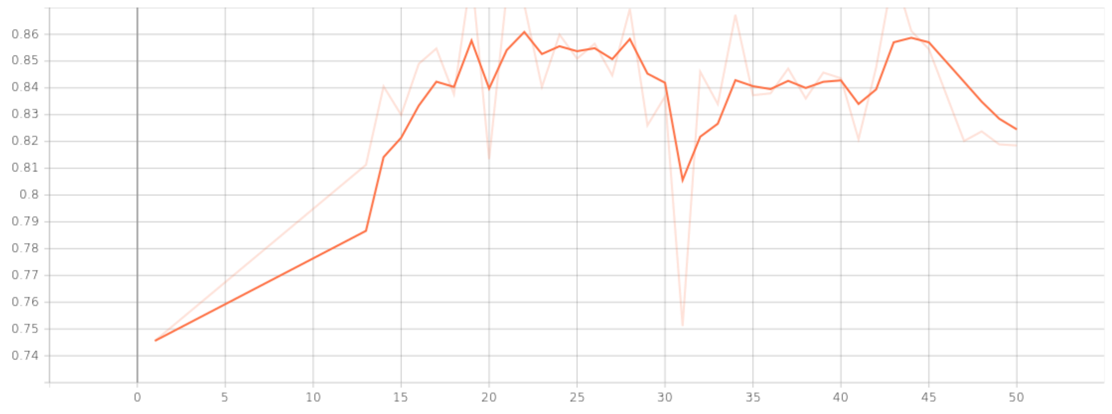

Figure 9, recall (Mask)

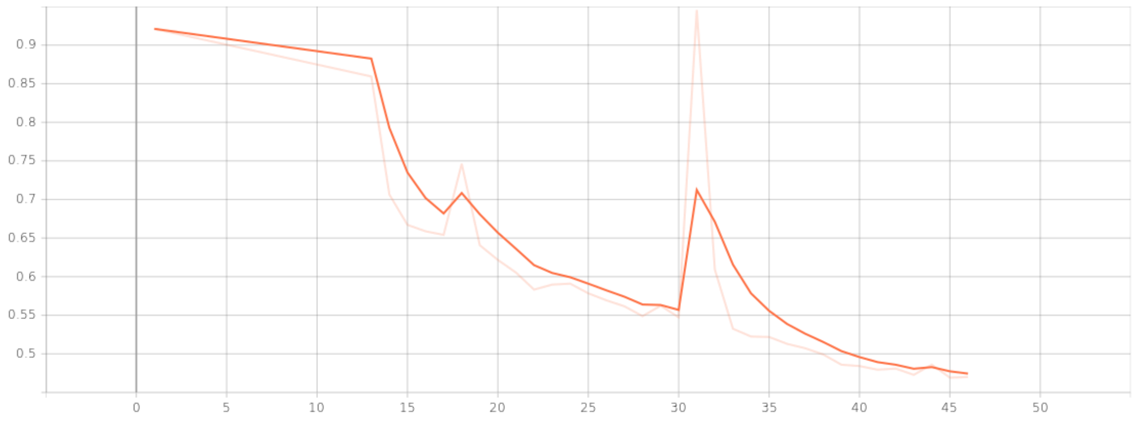

Figure 10, train box loss

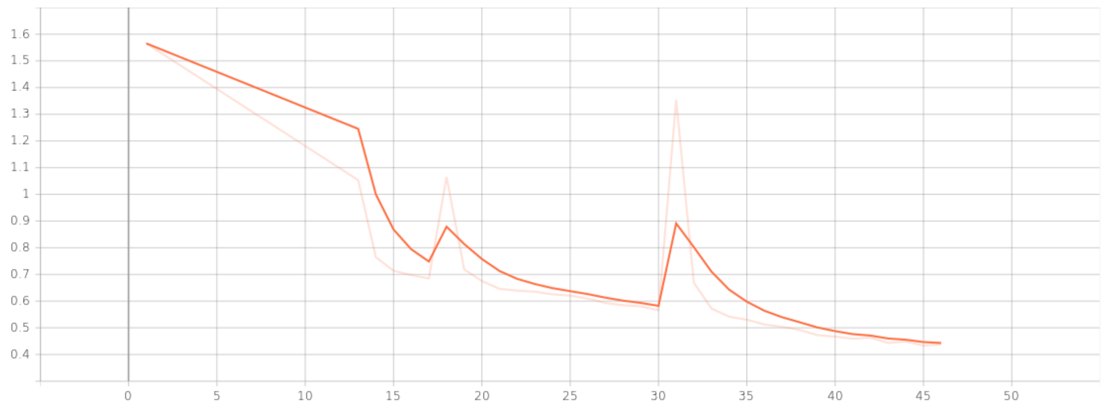

Figure 11, train class loss

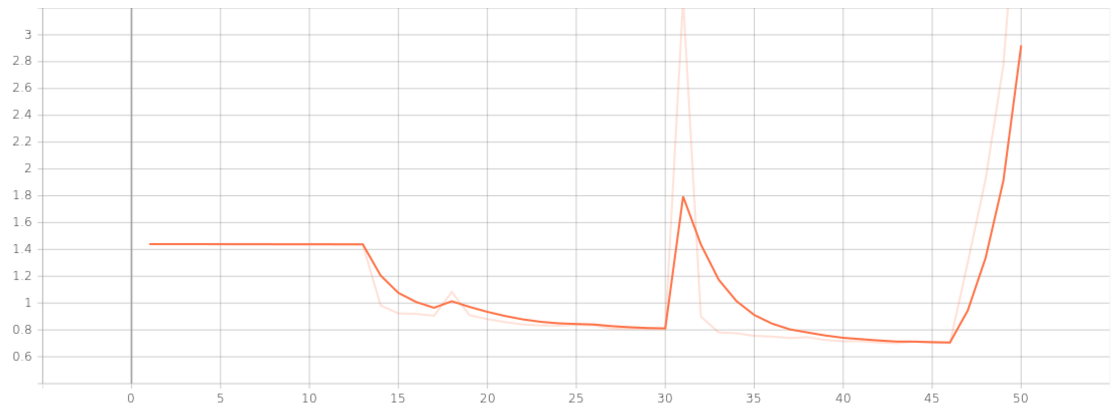

Figure 12, train segmentation loss

Validation:

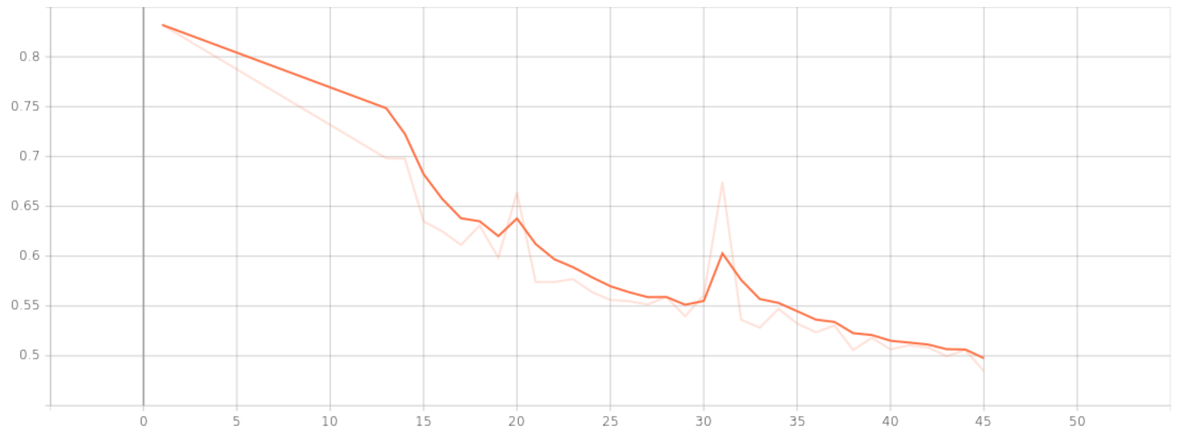

Figure 13, Val box loss

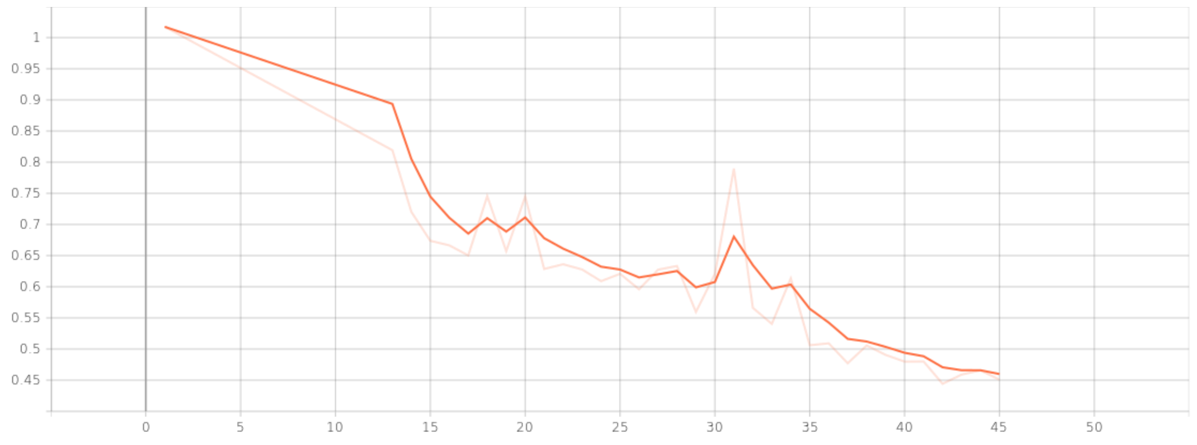

Figure 14, Val class loss

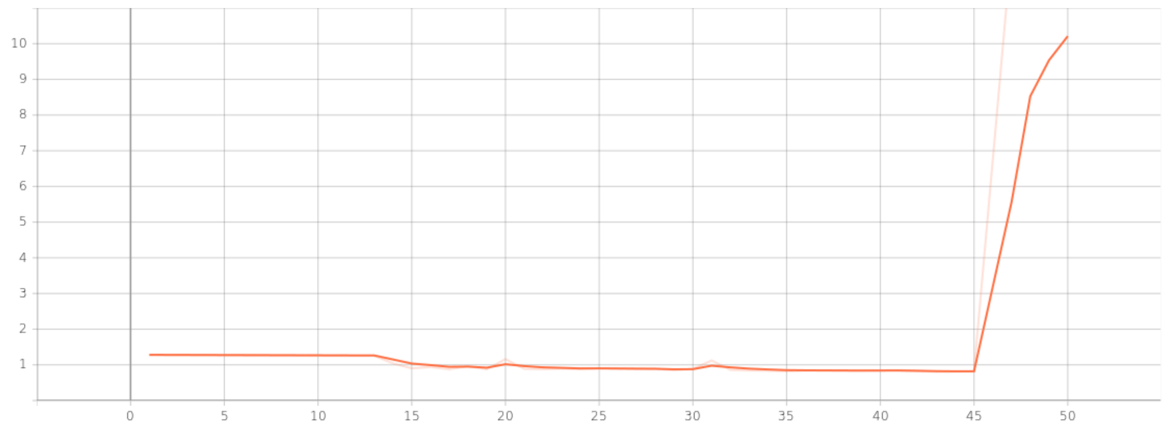

Figure 15, Val segmentation loss

As shown in figure 15 and 12, the model is over fitted after epoch 45. So, the best stage for model is in epoch 45.

Performance:

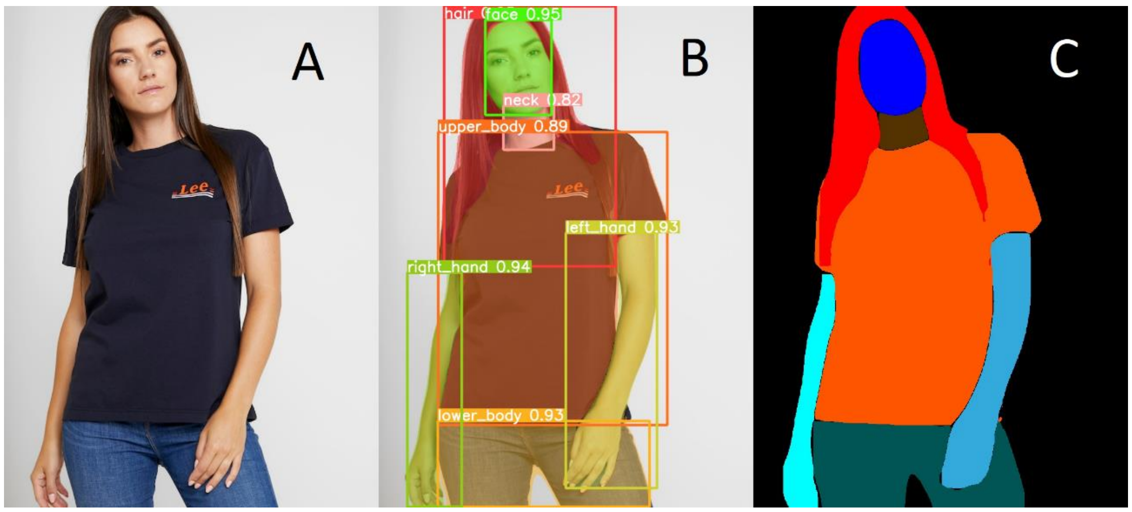

Figure 16, A) input image, B) segmented image C) created mask

Class	Images	Instances	Box Precision	Box Recall	Box mAP50	Box mAP50-95)	Mask Precision	Mask Recall	Mask mAP50	Mask mAP50-95)
All	200	1371	0.842	0.852	0.889	0.797	0.836	0.848	0.878	0.776
Hair	200	200	0.852	0.87	0.931	0.843	0.847	0.865	0.929	0.778
Neck	200	200	0.896	0.91	0.914	0.762	0.901	0.915	0.919	0.771
Upper body	200	194	0.899	0.918	0.921	0.871	0.904	0.923	0.931	0.879
Lower body	200	174	0.829	0.894	0.9	0.816	0.824	0.888	0.893	0.827
Left hand	200	189	0.816	0.757	0.864	0.757	0.821	0.762	0.875	0.742
Face	200	200	0.997	1	0.995	0.95	0.997	1	0.995	0.973
Right hand	200	192	0.718	0.865	0.878	0.763	0.721	0.87	0.883	0.746
Skirt	200	22	0.725	0.601	0.71	0.611	0.671	0.558	0.6	0.491

Table 1, YOLO-V8x segmentation performance

Speed: 7.5ms preprocess, 195.3ms inference, 14.0ms postprocess per image at shape (1, 3, 1280, 960)

Clothing-Agnostic Person Representation:

To train the model with pairs of clothing item c and image I where the person is already wearing cc, a person representation without clothing information in I is used for the virtual try-on task. Such representations must meet the following criteria:

1. the original clothing item to be replaced should be removed.
2. enough information to predict the person's pose and body shape should be retained.
3. regions to be preserved (e.g., face and hands) should be maintained to keep the person's identity intact.

We propose using a clothing-agnostic image $(I_a)$and a clothing-agnostic segmentation map (S_a) as inputs at each stage. These inputs effectively eliminate the clothing item's shape while preserving the body parts that need to be reproduced. First, we predict the segmentation map $(S \in L^{H\times W})$and the pose map $(P \in R^{3\times H\times W})$of the image (I) using pre-trained networks, where (L) is a set of integers representing semantic labels. The segmentation map (S) is used to remove the clothing region to be replaced and to preserve the rest of the image. The pose map (P) is utilized to remove the arms, but not the hands, as they are difficult to reproduce. Based on (S) and (P), we generate the clothing-agnostic image $(I_a)$and the clothing-agnostic segmentation map (S_a), which thoroughly remove the original clothing information while preserving the rest of the image. Additionally, unlike previous work that uses a pose heatmap with each channel corresponding to one key point, we concatenate (I_a)or $(S_a)$to the RGB pose map (P), representing a skeletal structure that improves generation quality.

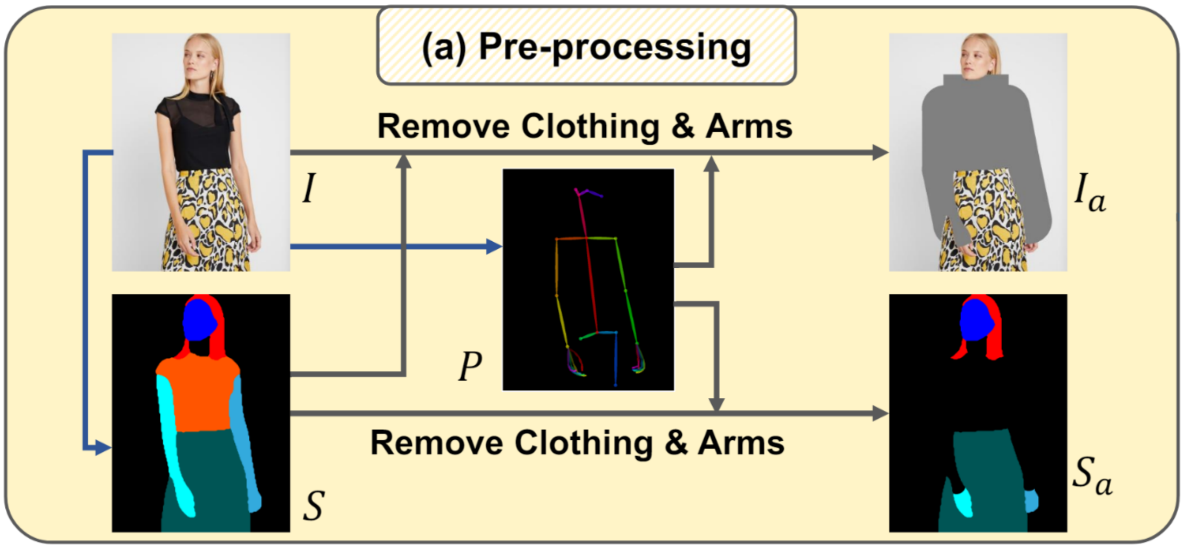

Figure 17, Clothing-Agnostic Person

Segmentation Generation:

Given the clothing-agnostic person representation $((S_a), (P))$ and the target clothing item (c), the segmentation generator (G_S)predicts the segmentation map $(\hat{S} \in L^{H\times W})$ of the person in the reference image wearing (c). We train (G_S) to learn the mapping between (S) and $((S_a), (P), (c))$, ensuring that the original clothing item information is completely removed. For the architecture of (G_S), we adopt U-Net [25]. The total loss (L_S) of the segmentation generator is defined as:

$$LS = LcGAN + \lambda CELCE$$

where (L_{CE}) and (L_{cGAN}) denote the pixel-wise cross-entropy loss and conditional adversarial loss between $(\hat{S})$ and (S), respectively. (λ_{CE}) is the hyperparameter that determines the relative importance between the two losses.

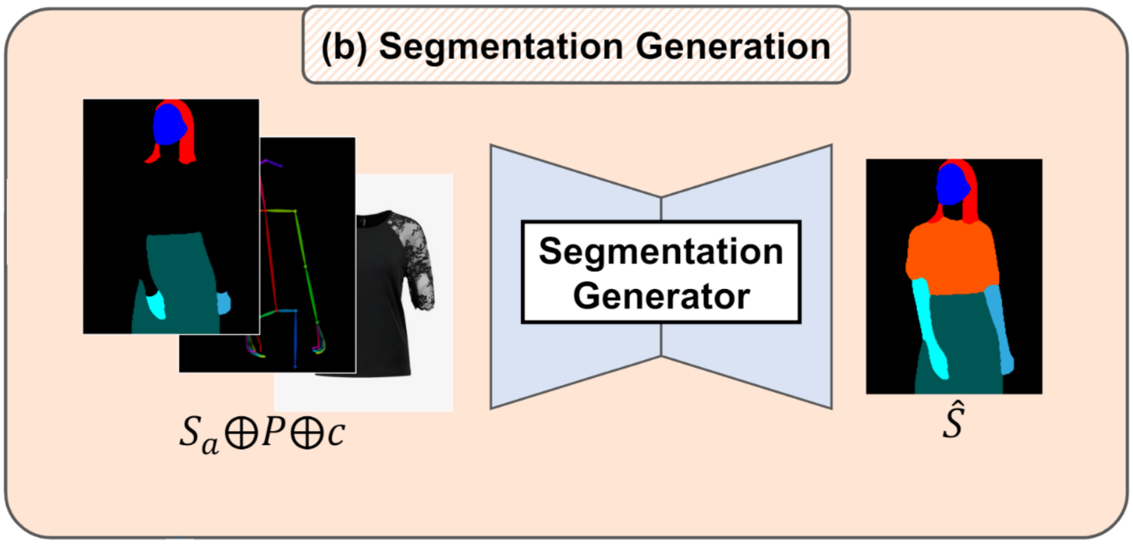

Figure 18, Segmentation Generation

Clothing Image Deformation

In this stage, we deform the target clothing item (c) to align it with $(\widehat{S_c})$, the clothing area of $(\hat{S})$. We employ the geometric matching module proposed in CP-VTON [31], using the clothing-agnostic person representation $((I_a), (P))$ and $(\widehat{S_c})$ as inputs. First, a correlation matrix is calculated between the features extracted from $((I_a), (P))$ and (c).

Using this correlation matrix, the regression network predicts the TPS transformation parameters ($\theta \in R^{2\times5\times5}$), $whicharethenusedtowarp(c)$. During the training phase, the model uses (S_c) extracted from (S) instead of ($\widehat{S_c}$). The module is trained using the L1 loss between the warped clothing and the clothing (I_c) extracted from (I). Additionally, a second-order difference constraint [35] is used to reduce distortions in the warped clothing images. The overall objective function for warping the clothes to fit the human body is given by:

$$[L_{warp} = |I_c - W(c, \theta)|_{1,1} + \lambda_{const} L_{const},]$$

where (W) is the function that deforms (c) using (θ), (L_{const}) is a second-order difference constraint, and (λ_{const})is the hyperparameter for (L_{const}).

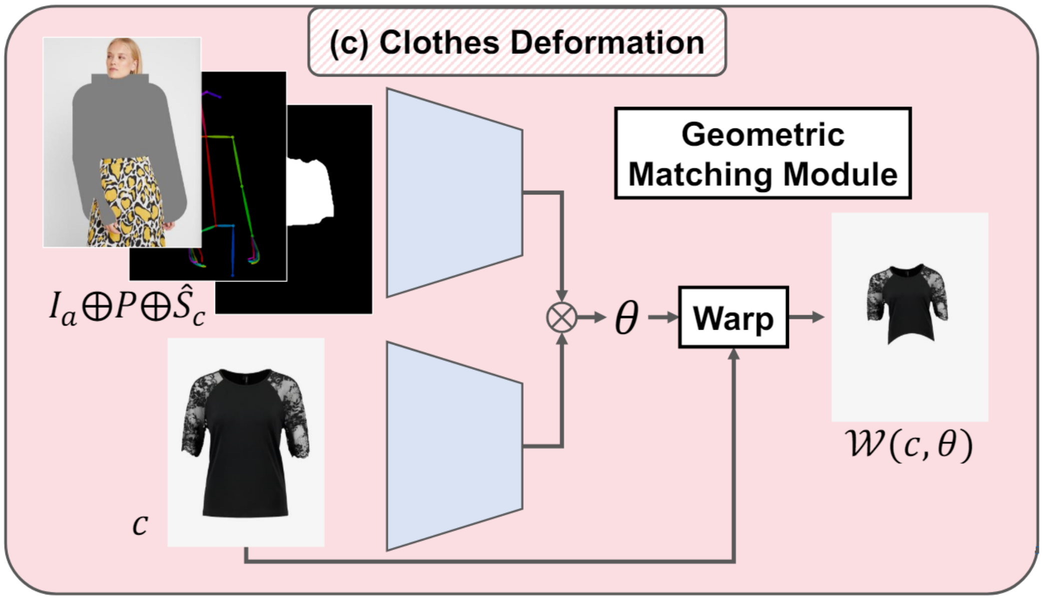

Figure 19, Clothing Image Deformation

Try-On Synthesis via ALIAS Normalization

Our goal is to generate the final synthetic image $(\hat{I})$based on the outputs from the previous stages. To achieve this, we fuse the clothing-agnostic person representation $((I_a), (P))$with the warped clothing image $(W(c, \theta))$, guided by $(\hat{S})$. The combination $((I_a), (P), (W(c, \theta)))$ is injected into each layer of the generator. For $(\hat{S})$, we introduce a new conditional normalization method called ALIAS (ALIgnment-Aware Segment) normalization. ALIAS normalization preserves semantic information and removes misleading information from misaligned regions by utilizing $(\hat{S})$ and the mask of these regions.

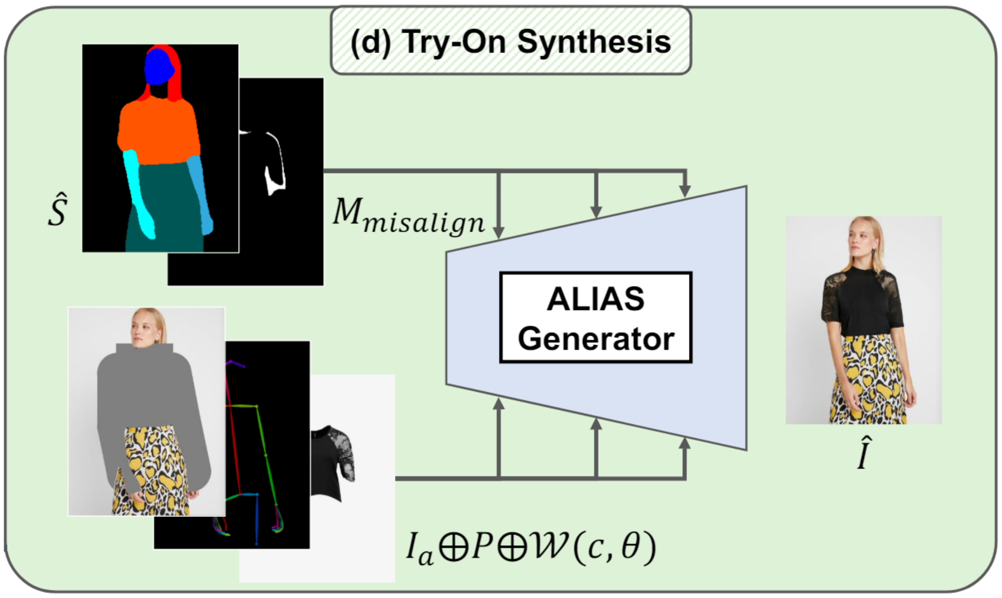

Figure 20, Try-On Synthesis via ALIAS Normalization