Transfer learning for Person Image with Attribute-Decomposed GAN

Haozhe Lei, Han Hao

Introduction and Problem Statement

The paper[1] that writen by Yifang Men's team already finished the transform from a certain clothes to other given human pictures by using Attribute-Decomposed GAN[2] (A-D GAN). In the meantime, it also works well on human poses transporting. However, does it still feasible for other data-set, such as unfashionable data sets, or not three-dimensional characters pictures, i.e., anime characters? This project will try to accomplish this unsolved issue.

Literature Survey

Processing an intact human picture is a tough work, since directly encodes the entire image might be a tedious task. However, Yifang Men adopted automatic and unsupervised component attributes generator[1] into the frame of GAN[2].

The goal of A-D GAN model is to synthesize high-quality person images with user-controlled human attributes, such as pose, head, upper clothes and pants. The corresponding keypoint-based can be automatically extracted via an existing pose estimation method[3].

For improving the generalization ability of texture encoding, inspired by a style transfer method[4] which directly extracts the image code via a pretrained VGG network, A-D GAN model introduce an architecture of global texture encoding by concatenating the VGG features in corresponding layers to its original encoder.

We use a newly discriminator method[5] in this model. This model adapts two discriminators D_p and D_t, and their specific attributes and responsibilities are explained in the discriminator sections.

And this model also introduce a new metric called contextual (CX) score, which is proposed for image transformation[6] and uses the cosine distance between deep features to measure the similarity of two non-aligned images, ignoring the spatial position of the features.

Description of the Dataset

We conduct experiments on the In-shop Clothes Retrieval Benchmark in the DeepFashion database. The dataset containing various poses and clothes, which is of large scale, diversities and quantities. It also has rich annotations, including 7982 number of clothing items.

In the original experiment design, the author randomly pick 101,996 pairs of images for training and 52,712 pairs for testing. We are going to reduce the scale of training to 1000 pairs for training and 75 pairs, because we make transfer learning from the pre-trained model, which could take advantage of the original large-scale model. We will crop images into 172 x 256 resolution which is $\{I \in R^{3\times172\times256}\}$, then we generate Component Transfer by human parsing model Look Into Person. Each image is segmented by 8 categories(i.e., background, hair, face, upper clothes, pants, skirt, arm and leg). As for Pose Transfer, we

will generate key points of body joints by using OpenPose, the output will be a 18 channel heatmap representing human pose $P \in R^{18\times172\times256}$ of I. Finally synthesis them together. Starting from the original dataset, the path of data processing is as shown in the following figure 1.

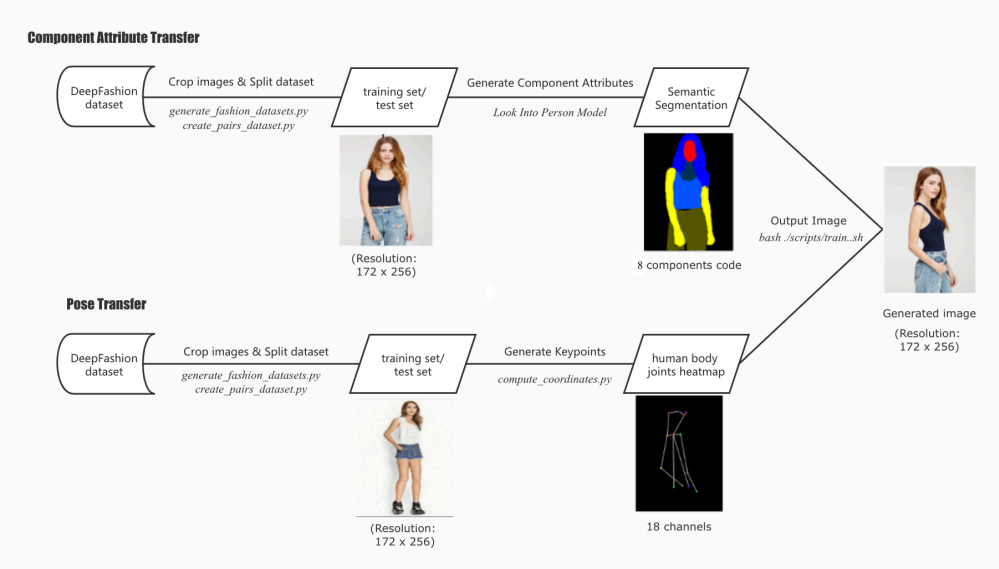

Figure 1: Data-Description

And when do the transfer learning, we use 14 3D-anime character pictures as the transfer learning dataset, pair by pair which means 7×7 data. Since this model is a really huge model and we only have limited resources, so the dataset could not set too big.

Description of the Model

This model divides the input human body pictures to multiply parts, and process them one by one. This processing not only faster the training time, but also utilize the strong correlation character of images, which means each parts could also enhance each other's abilities.

Generators: The input in figure 2 is a target pose P_t and source person image I_s, and the output is the generated image I_g with source person I_s in target pose P_t. This generator embeds P_t and I_s into two latent codes via two independent pathways, called pose encoding and decomposed component encoding, respectively.

The pose encoder is combined by a fixed VGG encoder and a learnable encoder by the method[4], which is called the global texture encoding (GTE) and could improve the performance of the model. The GTE module is shown in figure 3.

The source person image I_s will be embedded into the latent space as the style code via a module called decomposed component encoding (DCE) which is shown in figure 2. This DCE module decomposes the I_s into multiple components and recombines their latent codes to construct the full style code. This intuitive will: (1) First speed up the convergence of model and achieve more realistic results in less time. It can reduce the whole

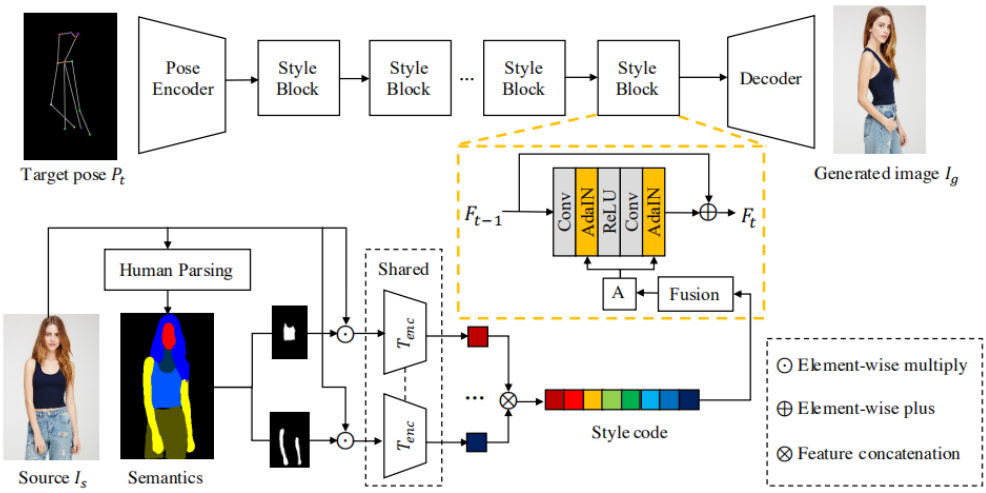

Figure 2: An overview of the network architecture of generator

person to different components, which could share the same network parameters for their encoder; (2) Second achieve an automatic and unsupervised attribute separation without any annotation in the training dataset, which utilizes an existing human parser for spatial decomposition. Specific attributes are learned in the fixed positions of the style code. Thus we can easily control component attributes by mixing desired component codes extracted from different source persons.

The transfer network consists of several cascaded style blocks, each one of which is constructed by a fusion module and residual conv-blocks equipped with AdaIN. This network could inject the texture pattern of source person into the feature of target pose, and the t^{th} style block is given by:

$$F_t = \varphi_t(F_{t-1}, A) + F_{t-1} \tag{1}$$

Where F_t and F_{t-1} are deep features, the first block is $F_0 = C_{pose}$, φ_t is a conv-blocks and A denotes learned affine transform parameters.

The person image reconstruction will be done by decoder generates using final target features F_{T-1} at the output of the last style block via N deconvolutional layers, following the regular decoder configuration.

Discriminators: Following Zhu et al.[5], this model adapts two discriminators D_p and D_t, where D_p is used to guarantee the alignment of the pose of generated image I_g with the target pose P_t, and D_t is used to ensure the similarity of the appearance texture of I_g with the source person I_s. For D_p, the target pose P_t concatenated with the generated image I_g (real target image I_t) is fed into D_p as a fake (real) pair. For D_t, the source person image Is concatenated with Ig (I_t) is fed into D_t as a fake (real) pair.

Description of the Loss Function

Our full training loss is composed by 4 terms, a reconstruction term, a perceptual term and a contextual term.

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{per}\mathcal{L}_{per} + \lambda_{CX}\mathcal{L}_{CX} \tag{2}$$

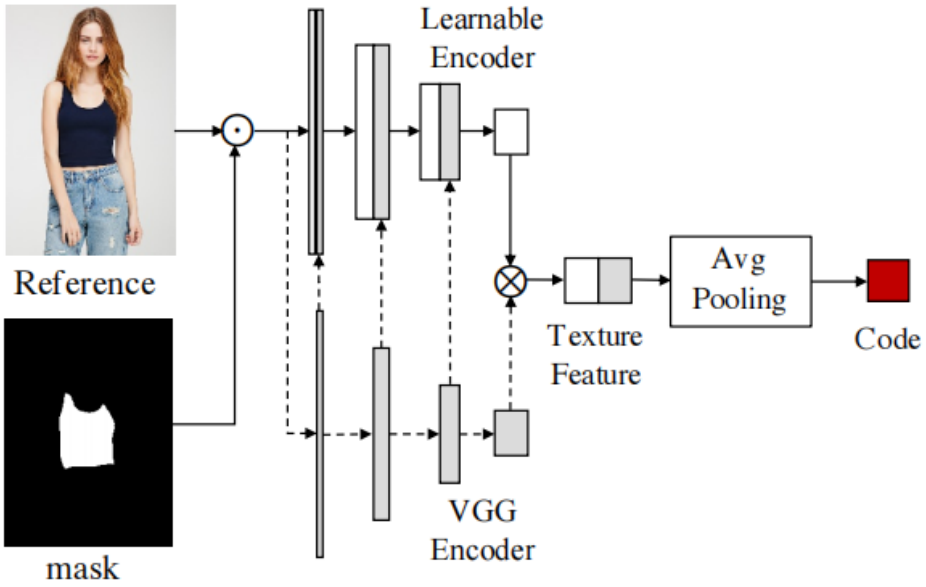

Figure 3: VGG Transfer Method

where $\lambda_{rec}, \lambda_{per}$ and λ_{CX} denote the weights of corresponding losses, respectively.

$\mathcal{L}_{adv}$ is the adversarial loss with discriminators D_p and D_t for helping the generator G synthesize the target person image, which will be given by:

$$\begin{aligned}\mathcal{L}_{adv} = & E_{I_s,P_t,I_t}[log(D_t(I_s, I_t) \cdot D_p(P_t, I_t))] \\ & + E_{I_s,P_t}[log((1 - D_t(I_s, G(I_s, P_t))) \cdot (1 - D_p(P_t, G(I_s, P_t))))]\end{aligned} \tag{3}$$

$\mathcal{L}_{rec}$ is the reconstruction loss used to directly guide the visual appearance of the generated image similar to the target image I_t, which will be given by:

$$\mathcal{L}_{rec} = \|G(I_s, P_t) - I_t\|_1 \tag{4}$$

$\mathcal{L}_{per}$ is the perceptual loss which is used to exploit deep features extracted from certain layers of the pretrained VGG network for texture matching, which will be given by:

$$\mathcal{L}_{per} = \frac{1}{W_l H_l C_l} \sum_{x=1}^{W_l} \sum_{y=1}^{H_l} \sum_{z=1}^{C_l} \|\phi_l(I_g)_{x,y,z} - \phi_l(I_t)_{x,y,z}\|_1 \tag{5}$$

In it, ϕ_l is the output feature from l of VGG19 and W_l, H_l,C_l are spatial width, height and depth of feature ϕ_l.

After create the model, the output should be the synthesised picture generated with editable style codes from the inputs.

Finally, we will use CX score[6], which is able to explicitly assess the texture coherence between two images and it is suitable for our task to measure the appearance consistency between the generated image and source image (target image), recording as CXGS (CX-GT). The loss function will be computed by:

$$\mathcal{L}_{CX} = -log(CX(\mathcal{F}^l(I_g), \mathcal{F}^l(I_t))) \tag{6}$$

Where $\mathcal{F}^l(I_g)$ and $\mathcal{F}^l(I_t)$ denote the feature maps extracted from layer l.

Implementation Details

Our method is implemented in Pytorch 1.8.1+cu101 using one NVIDIA Tesla P100-PCIE-16GB. The weights for the loss terms are set to $\lambda_{rec} = 2$, $\lambda_{per} = 2$, and $\lambda_{CX} = 0.02$. This model adopts Adam optimizer with the momentum set to 0.5 to train our model for around 120k iterations with learning rate equals $1e^{-3}$. The initial learning rate is set to 0.001 and linearly decayed to 0 after 60k iterations.The Hyperparameter selection sheet is shown in figure 4.

Parameter	Value	Meaning
batchSize	1	Input batch size
imgSize	176x256	Crop images to this size
input_nc	3	Number of input channels
output_nc	3	Number of output channels
ngf	64	Number of gen filters in the first convolution layer
ndf	64	Number of discrim filters in the first convolution layer
which_model_netD	Resnet	Set model to use for netD
which_model_netG	PATN	Set model to use for netG
G_n_downsampling	2	Down-sampling blocks for generator
D_n_downsampling	2	Down-sampling blocks for discriminator

Figure 4: Hyperparameter Sheet

Project Outcomes

The results are shown below, in the figure 5. We see the first row's first picture, this model still works well when source and target images are truly human beings. That shows the transfer learning does not change the change the existing capabilities of the model. And in the second rows, we could find the result image which take anime characters as source or target images, but the other uses pictures of truly human beings. We can see, if we use anime characters in target image, the result image will perform better compare

with we use it in source image. That is because the target image is more focus on the pose while the source image is more focus on the texture or style of the image. And the training of texture or style is harder than pose, the former, after all, is more complex. When look back to the first row's second picture, the same conclusion can be said for case we use anime character pictures in both source and target images. We can find that the posture of the character is well restored, but the clothing is a little bit wrong in this situation.

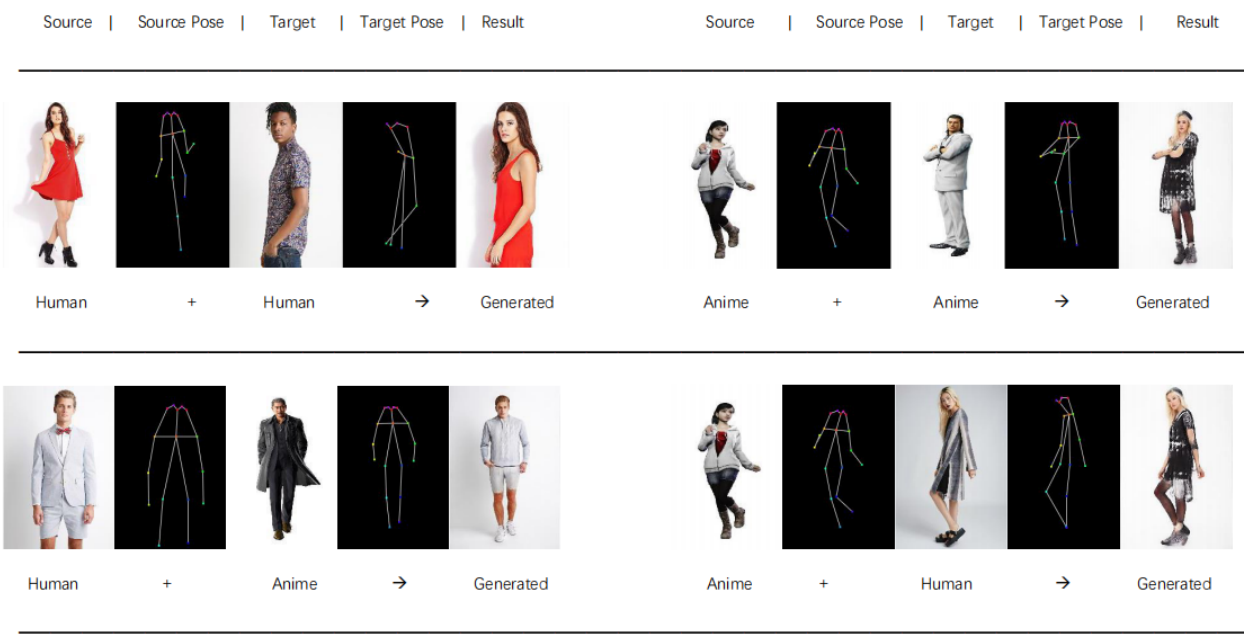

Figure 5: Transform style result within anime characters and truly human beings

Plot of Learning Rate

We use 14 3D-anime character pictures pair by pair to do the transfer leaning, and the result is shown below in 6. We could find the two discriminator losses keep reducing,

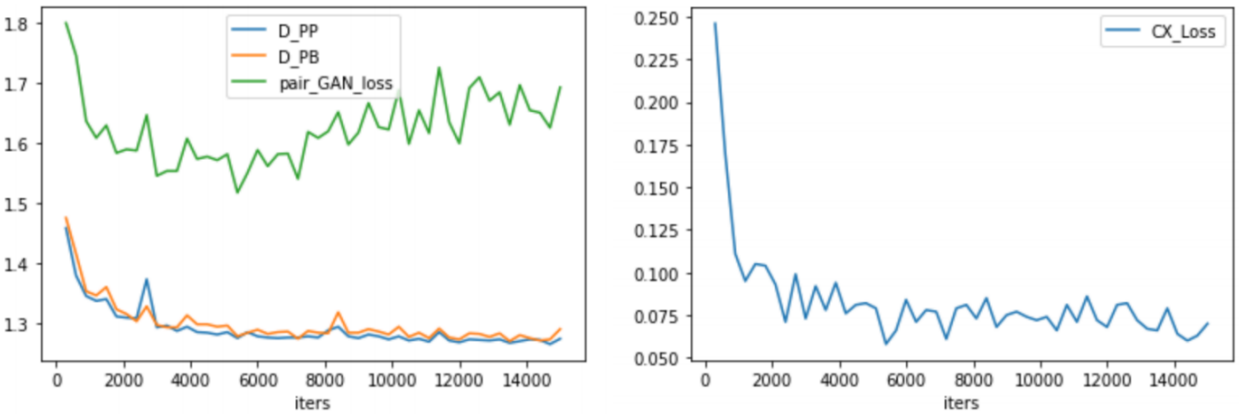

Figure 6: Plot of training loss

and the loss of generator first decrease than increase. The reason might be the lacking of the dataset, and existing strength of the discriminator. In this situation, the loss of Discriminator is relatively easier to train, therefore it grows too powerful. And the generator will tend to increasingly overstretched in faking the discriminator. However, the CX loss looks really well, because the generator, after all, is always evolving. So the result of this model is relatively reasonable.

Conclusion Section

What we intend to do: We hope in this project we could achieve following tasks: (1) Achieve component attributes transform model to deepfashion dataset; (2) Transform this model to anime dataset and evaluate the result; (3) Finally create a model which could transform fashion style and pose between different anime characters.

What we have accomplished: We built the A-D GAN model and work it fine, with the correct set of outputs with deepfashion dataset. And we transform this model to anime dataset. We think the reasult barely satisfactory, since we did not have a too big dataset and we did not use too many training epoch. The huge model structure leads to a really long training time and really big resource consumption. We were hoping to use a 2D-anime character, but it was too difficult for the discriminator, so we used a 3D-anime character instead. We felt that with a better machine and more time, the model would perform better and could transform to 2D-anime character pictures.

Future prospects: This model is a really powerful model, we believe it has the ability to process anime characters image transform. Today's social game to the artist occupation requirements are increasing day by day, if we could give them the tool which could provide character design reference conveniently, that will save a lot of manpower and material resources.

However, this network is so large that it is a bit redundant, and training this network would be not only time-consuming, but also extremely power-consuming. Therefore, we may need to find some solution to reduce the training cost of this network, whether with new network architecture, or with special data pre-processing methods.

Publicly accessible link

Google Colab Google Drive GitHub

References

[1] Yifang Men et al. "Controllable Person Image Synthesis with Attribute-Decomposed GAN". In: *Computer Vision and Pattern Recognition (CVPR), 2020 IEEE Conference on*. 2020.

[2] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[3] Zhe Cao et al. *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. 2019. arXiv: 1812.08008 [cs.CV].

[4] Xun Huang and Serge Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. 2017. arXiv: 1703.06868 [cs.CV].

[5] Zhen Zhu et al. “Progressive Pose Attention Transfer for Person Image Generation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[6] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. *The Contextual Loss for Image Transformation with Non-Aligned Data*. 2018. arXiv: 1803.02077 [cs.CV].