Fashion Fitting in Deep Learning

Hui Kwat Kong
20123133

Song Yangyang
20534320

Abstract

"Fashion Fitting", is to transfer an arbitrary clothes from the model to the target person, where two steps of model have been developed: (i) segmentation and fitting. Segmentation applies the blend of image-processing technique (Grabcut) and Deep Learning algorithm (CNN). (ii) The fitting part (fit the segmented clothing to the target person) employs Conditional Analogy GAN (CAGAN) architecture. The final model is able to realize the "Fashion Fitting" function.

1. Introduction

With the fast-paced lifestyle and rapid development of the internet technology, online shopping is more preferred than the traditional shopping physically. However, it is difficult to predict how the clothes fits the person without seeing the whole image of wearing the particular clothes. Hence, this project analyzes the state-of-the-art model using DeepFashion dataset, pose transfer technologies to transfer arbitrary clothing to a persons clothing.

2. Related work

2.1. M2E-Try On Net

The virtual Try-On network, M2E-Try On Net[1], transfers the clothes from a model image to a person image without the need of any clean product images, which is able to align the poses between the model and the target person with preserving the models clothes.

2.2. SwapNet

SwapNet[3] is able to swap the clothing between a pair of images while preserving the pose and body shape. There are two stages including warping stage and texturing stage.

2.3. A Variational U-Net for Conditional Appearance and Shape Generation

U-Net[1] is used for mapping from shapes to target images and for conditioning potential representations of variational autoencoders about appearance.

3. Dataset

DeepFashion[2] contains over 800,000 diverse fashion images ranging from well-posed shop images to unconstrained consumer photos. Each image in this dataset is labeled with 50 categories, 1000 descriptive attributes, bounding box and clothing landmarks. The dataset also contains over 300,000 cross-pose/cross-domain image pairs.

In-shop Clothes Retrieval Benchmark: It evaluates the performance of in-shop Clothes Retrieval. The dataset contains large pose and scale variations for each clothing. However, it is very difficult to transform some clothing (even segment them out of the person), for example: All the lower body part (demin, pants, leggings) and most of the clothing for women are ignored. It will be discussed in section 5 at page 5.

Figure 1. women clothes and men clothes

3.1. Model Training

We analyze the state-of-the-art model using DeepFashion dataset and related technologies to transfer arbitrary models clothes to a target persons clothes. The whole process is to extract the target persons clothes and the models clothes and then developing a model to realize clothes fitting.

It can be formulated as follow: given a target person with clothing x_i and a model person with clothing x_j, we generate the corresponding segmented clothing y_i and y_j respectively, then generate the target person image but with model

clothing, denoted as $G(x_i, y_i, y_j)$. The idea is illustrated in figure 2.

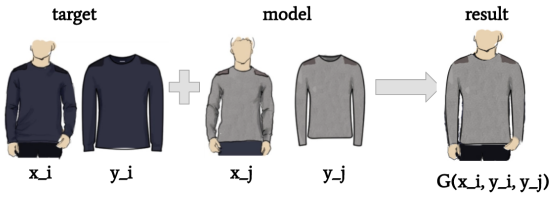

Figure 2. the whole process of fashion fitting

The pipeline of model training includes two steps: (i) clothes segmentation and (ii) fitting.

3.2. Step1: Segmentation

By using the blend of image-processing technique (Grabcut) and Deep Learning algorithm (CNN), we can segment the clothes texture from images.

3.2.1 Grabcut

Grabcut is an image segmentation method based on graph cuts which can use opencv to get the library.

The key parameters used in Grabcut are:

a. img:the original image

b. mask: a mask for specific shape

c. rectangle: four coordinates (start_x, start_y, width, height) to locate a specific rectangle which is treated as the bounding box

d. bgdModel: background models with zero array in fixed shape of (1,65)

e. fgdModel: foreground models with zero array in fixed shape of (1,65)

f. iteration n: iteration count

g. cv2.GC_INIT_WITH_RECT: one of the grabcut mode which uses the reactangle

After computing the new mask, multiplying the original image and the new mask can give the good segmentation of clothes.

3.2.2 Segmentation

a. Starting with a user-specified bounding box around the clothes to be segmented, the algorithm estimates the color distribution of the target clothes and that of the background using a Gaussian mixture model.

b. Then construct a Markov random field over the pixel labels, with an energy function that prefers connected regions having the same label, and running a graph cut based optimization to infer their values.

c. This two-step procedure is repeated until convergence.

3.2.3 Results

With the usage of segmentation with Grabcut, the model is well enough to segment images in different backgrounds. It can be seen some results below which not only test the arbitrary images but also the DeepFashion dataset we used in this project.

a. some results in arbitrary images:

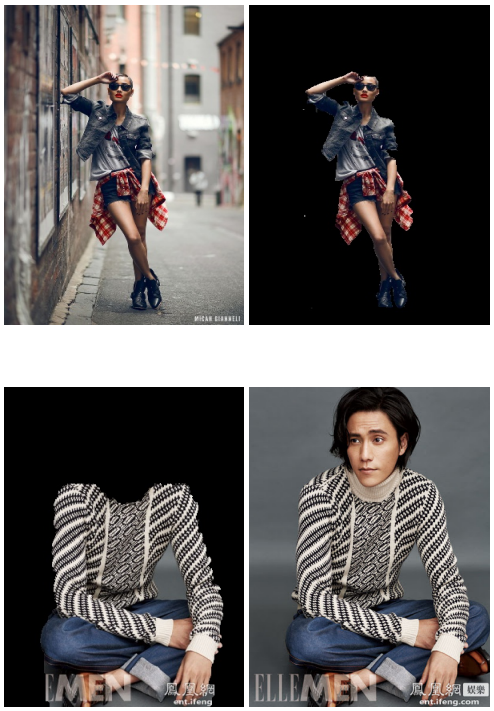

Figure 3. segmentation of arbitrary images

Noted that in the two examples (Figure 3), the body part (legs and hands) cannot be completely separated. These examples (for DeepFashion dataset) are ignored in the next step.

b. better results using DeepFashion dataset: (Figure 4)

3.3. Step2: Fitting

Then the state-of-the-art deep learning architecture, Conditional Analogy Generative Adversarial Network (CA-GAN, one type of GAN structures) is applied to fit a given clothes into the target person.

Figure 4. segmentation of DeepFashion dataset images

3.3.1 CAGAN(Conditional Analogy GAN)

CAGAN [4] is able to learn the relation between paired images present in training data, and then generalize and generate images that correspond to the relation, but they were never seen in the training dataset. We adapt this adversarial training and deep convolutional neural network (CNN) to automatically swap the clothing on fashion model photos, with Deep Fashion dataset of original images and segmentation images (from last step) as training data.
Training of the GAN model is to learn a generator G to generate plausible images which fool a discriminator D. The discriminator D needs to distinguish that whether the image is reasonable or not and whether the clothes is well-fitted in the target persons body or not.

3.3.1.1 Generator

- Input: 500x300 pixel images from our output images from step 1.
- Output: The output of generator is four channels: the mask and 3 RGB channels with following formula:

$$x_i, y_i, y_j \mapsto [\alpha_{ij}, (x_{ij}^R, x_{ij}^G, x_{ij}^B)] \mapsto x_{ij} \qquad (1)$$

where

$$x_{ij} = \alpha_{ij} \times (x_{ij}^R, x_{ij}^G, x_{ij}^B) + (1 - \alpha_{ij}) x_i \qquad (2)$$

The output image (x_{ij}^R, x_{ij}^G, x_{ij}^B) can be regarded as the summation of 3 RGB channels with agreed by the mask α_{ij} (like an indicator function) and the original image x_i with disagreed by the mask (indicator function).

- Structure of the generator

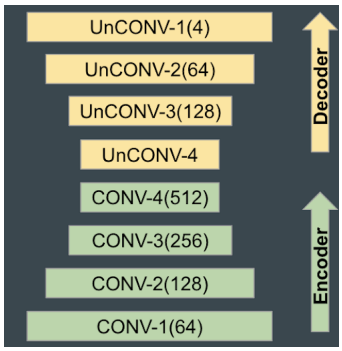

Figure 5. structure of generator

This is a typical UNET structure with down-sampling (green part) and up-sampling (yellow part) with transposed Convolution layers. There are concatenations from encoder layer directly to the decoder network. Noted that instance normalization has been chosen instead of batch normalization because of its better performance and design for GAN task.[5]

3.3.1.2 Discriminator

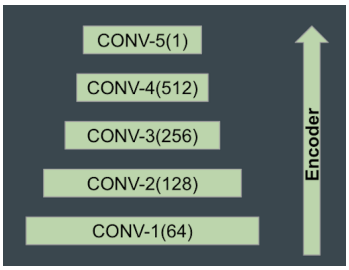

Figure 6. structure of discriminator

The discriminator is a standard down-sampling CNN structure with output as 0 or 1. Here, the batch normalization is applied here.

3.3.1.3 Loss function

The loss is divided into three parts: The classical GAN loss, the L1-regularization and the cycle loss in the training process. The whole optimization of the loss function is similar to classical GAN structure:

$$\min_G \max_D \left(L_{GAN}(G, D) + L_{id}(G) + L_{cyc}(G) \right) \qquad (3)$$

- GAN loss The GAN loss (denoted as $L_{GAN}(G, D)$) is again the summation of 3 parts:

$$L_{GAN}(G,D) = \mathbb{E}_{x_i,y_i\sim p_{\text{data}}} \sum_{\lambda,\mu}[\log D_{\lambda,\mu}(x_i,y_i)] + \mathbb{E}_{x_i,y_i,y_j\sim p_{\text{data}}} \sum_{\lambda,\mu}[(1-\log D_{\lambda,\mu}(G(x_i,y_i,y_j),y_j))] + \mathbb{E}_{x_i,y_{j\neq i}\sim p_{\text{data}}} \sum_{\lambda,\mu}[(1-\log D_{\lambda,\mu}(x_i,y_j))] \quad (4)$$

Noted $x_i, y_i \sim p_{data}$ means it is randomly selected from uniform distribution. There the first and second terms are the classical GAN loss, the last term is to make the discriminator more robust that the direct matching without generation is not legitimate. The idea is illustrated in below:

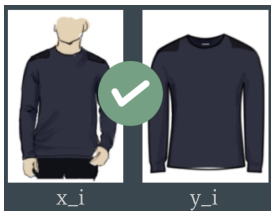

Figure 7. First term loss

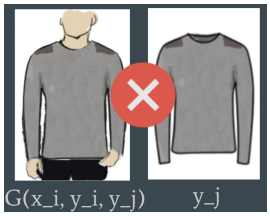

Figure 8. Second term loss

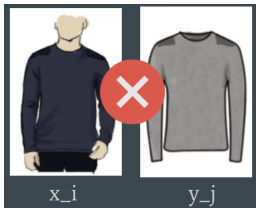

Figure 9. Third term loss

- L1-regularization
  Basically, one can treat the regularization is to make generating image change as less as possible from the original target person image x_i.

- Cycle loss
  This loss (denoted as $L_{cyc}(G)$) is to stabilize the generated image. Basically, the twice swapping image of x_i (G(G(x_i, y_i, y_j), y_j, y_i)) needs to be as close as x_i as possible.

3.3.2 Environment set-up

The model is trained at Ubuntu Linux AWS (Amazon Web Service) EC2 instance: p3.2xlarge, which includes 1 NVIDIA Tesla V100 GPUs with 16GB GPU memory and 8 vCPUs with vCPUs. To connect the code in local and download the result, github and cyberduck are applied here.

3.3.3 Training data

The training is based on 46,872 swapping case of images. (217 clothing images with successful segmentation from step 1) Noted that the number is restricted due to some failure cases in segmenting the clothes image in step 1 and the same limitation in section 5.

3.3.4 Hyperparameter setting

There are many hyperparameters involved in generator, Discriminator and loss functions (L1 or L2 and the weighting of regularization and cycle loss). Due to the limited budget in using AWS EC2 instance and based on our purpose of this project, we instead preset some reasonable hyperparameters in short. Some of them are listed below:

a. Convolutional layer

- kernel: 4
- stride: 2

b. Normalization

- generator: instance normalization
- discriminator: batch normalization

c. Activation function

- hidden layer: leaky ReLU
- last layer: sigmoid and tanh

d. Optimizer

- Adam
- learning rate: 2e-4
- beta$_1$: 0.5

4. Results

By training GAN for thousand of iterations, the results (snapshot from the model over specific number of iteration) are different and getting better and better aftering more times of iterations.

- After 1-iteration:

The clothes are almost not changed from target person.

- After 4000-iteration:

Figure 10. the result of 1-iteration

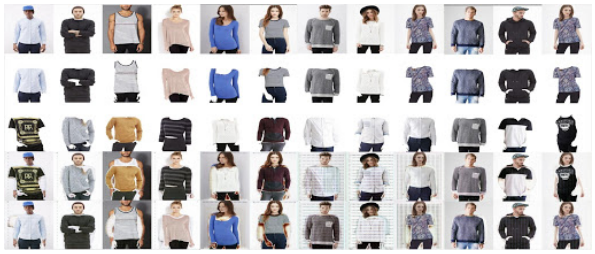

Figure 11. the result of 4000-iteration

Figure 12. the result of 9000-iteration

The model clothes can fit into the target person but still not find the best place to fit the clothes in. There are a lot of errors and skewing here.

- After 9000-iteration:

The results are relatively better than 4000-iterations except some troubles, such as identifying girls hair and specific styles.

5. Limitations

There are few limitations in this project:

- Data issue:
  It is very difficult to transform some clothing (even segment them out of the person), for example: we ignore the lower body part (demin, pants, leggings) and most of the clothing for women.

- Computing capability:
  GAN heavily requiures computing power and GPU. In our case, training a model in AWS EC2 instance p3.2xlarge is expensive.

- Pose transfer is still a very popular research topic that is not sophisticated (in production level) enough to be implemented.

- Women is very different and difficult in terms of fashion style and their long hairs which covers the clothes.

- The position (inclined to the left or right) and scaling are different among the photos.

6. Future works

To solve the last limitation, it is suggested that one can apply the mask generated from part1 to transform inclined image to the centre, such as finding the x-axis midpoint of the mask.

Figure 13. Inclined person to centre

References

[1] Liu Z, Luo P, Qiu S, et al. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1096-1104.

[2] Esser P, Sutter E, Ommer B. A variational unet for conditional appearance and shape generation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8857-8866.

[3] Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, Jingwan Lu. SwapNet: Image Based Garment Transfer. ECCV (12) 2018: 679-695

[4] Jetchev N , Bergmann U . The Conditional Analogy GAN: Swapping Fashion Articles on People Images[J]. 2017.

[5] K. (2018, November 30). An Overview of Normalization Methods in Deep Learning. Retrieved May 8, 2019, from http://mlexplained.com/2018/11/30/an-overview-of-normalization-methods-in-deep-learning/