Fashion Fitting in Deep Learning

Group 2
Hui Kwat Kong 20123133
Song Yangyang 20534320

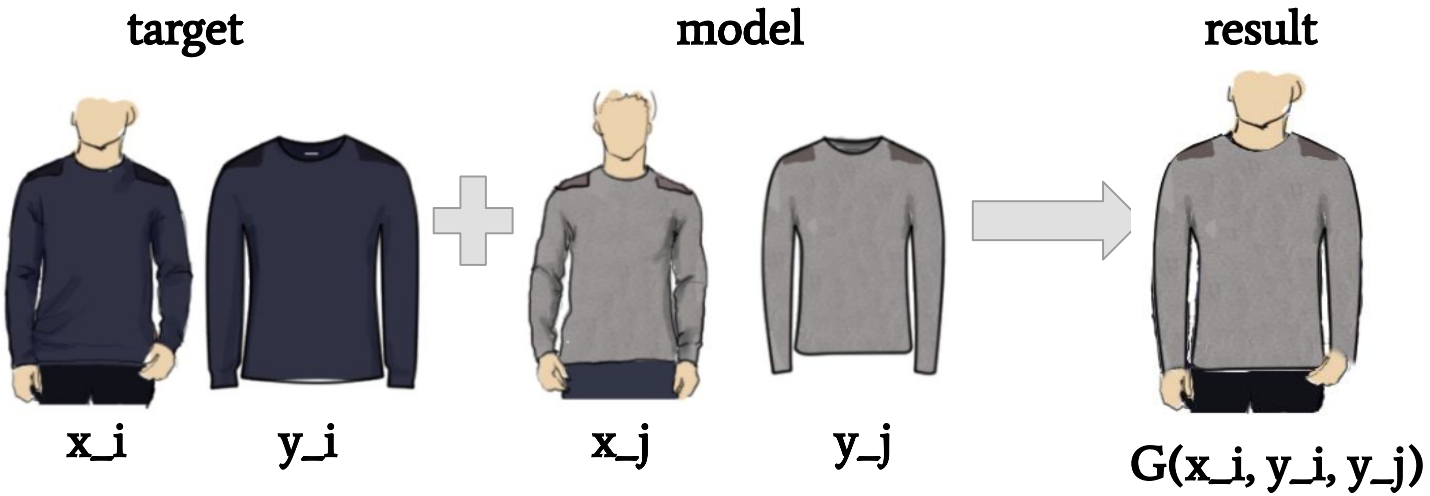
Content

1. Introduction
2. Dataset
3. Step1: segmentation
4. Step2: Conditional Analogy GAN
5. Results
6. Limitations & Future work

Introduction

With the rapid change in online shopping lifestyle including fashion, how do you know the clothes really fit you in online?

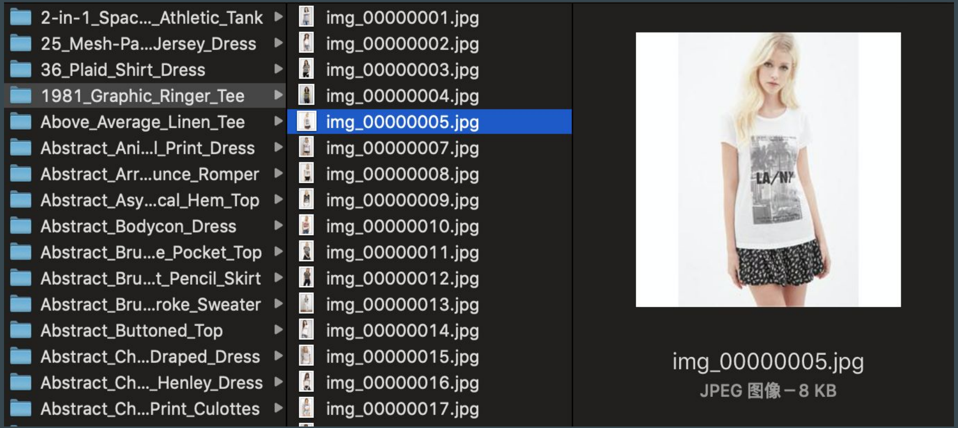
SO... We want to analyze the state-of-the-art model using DeepFashion dataset and related technologies to transfer arbitrary model's clothes to a target person's clothes.



Datasets

DeepFashion 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.

“In-shop Clothes Retrieval Benchmark”: It evaluates the performance of in-shop Clothes Retrieval. The dataset contains large pose and scale variations for each cloth.



Pipeline of model training

2 steps

- step 1: clothes segmentation from model (segmentation)
- step 2: fit clothes into target person (transfer learning in CAGAN)

Pipeline of model training - step1: segmentation

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

Grabcut: an image segmentation method based on graph cuts

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

Segmentation- grabCut

cv2.grabCut		
(img,	→	Load original images
mask,	→	Create a mask for specific shape
rect,	→	(start_x, start_y, width, height)
bgdModel,	→	Specify Background models(1,65)
fgdModel,	→	Specify Foreground models(1,65)
n,	→	Iteration Count
cv2.GC_INIT_WITH_RECT)	→	Mode, which uses rectangle

Pipeline of model training - step1: segmentation

some results in arbitrary images:

good results using Deep Fashion dataset:

Pipeline of model training - step2: fitting

By using GAN (Generative adversarial network) and Transfer Learning of CAGAN (Conditional Analogy GAN), we can train a new GAN to fit given clothes into target person.

CAGAN allows to learn the relation between paired images present in training data, and then generalize and generate images that correspond to the relation. As it is based on adversarial training and employs deep convolutional neural networks. An especially interesting application of that technique is automatic swapping of clothing on fashion model photos. We use Deep Fashion dataset of original images and segmentation images as training data, aftering a long time training, a model is built and can be used in new images for fitting clothes.

Conditional Analogy GAN (Adam + learn rate= 2e-4)

Generator

(Leaky ReLU, ReLU, Concatnate & InstanceNorm)

UnCONV-1(4)

UnCONV-2(64)

UnCONV-3(128)

UnCONV-4

CONV-4(512)

CONV-3(256)

CONV-2(128)

CONV-1(64)

Decoder

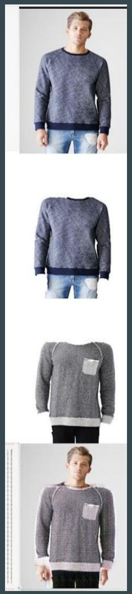
Encoder

α_i^j mask

$\tilde{x}_i^j$ 3 RGB channels

$$x_i, y_i, y_j \mapsto \left[\alpha_i^j, \tilde{x}_i^j\right] \mapsto x_i^j$$

$$x_i^j = \alpha_i^j \tilde{x}_i^j + \left(1 - \alpha_i^j\right) x_i$$



x_i

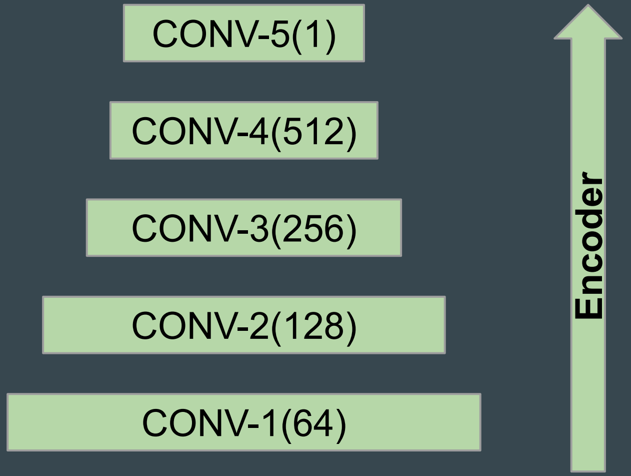
y_i

y_j

G(x_i, y_i, y_j)

Conditional Analogy GAN (Adam + learn rate= 2e-4)

Discriminator(LeakyReLU & BatchNorm)



Output of Discriminator: 0, 1

Conditional Analogy GAN (loss)

I. GAN loss

If you were Discriminator:

x_i y_i G(x_i, y_i, y_j) y_j x_i y_j

Conditional Analogy GAN (loss)

I. L1-Regularization (||·|| is L1 loss)

$$\mathcal{L}_{id}(G) = \mathbb{E}_{x_i, y_i, y_j \sim p_{\text{data}}} \left\| \alpha_i^j \right\|$$

II. Cycle loss

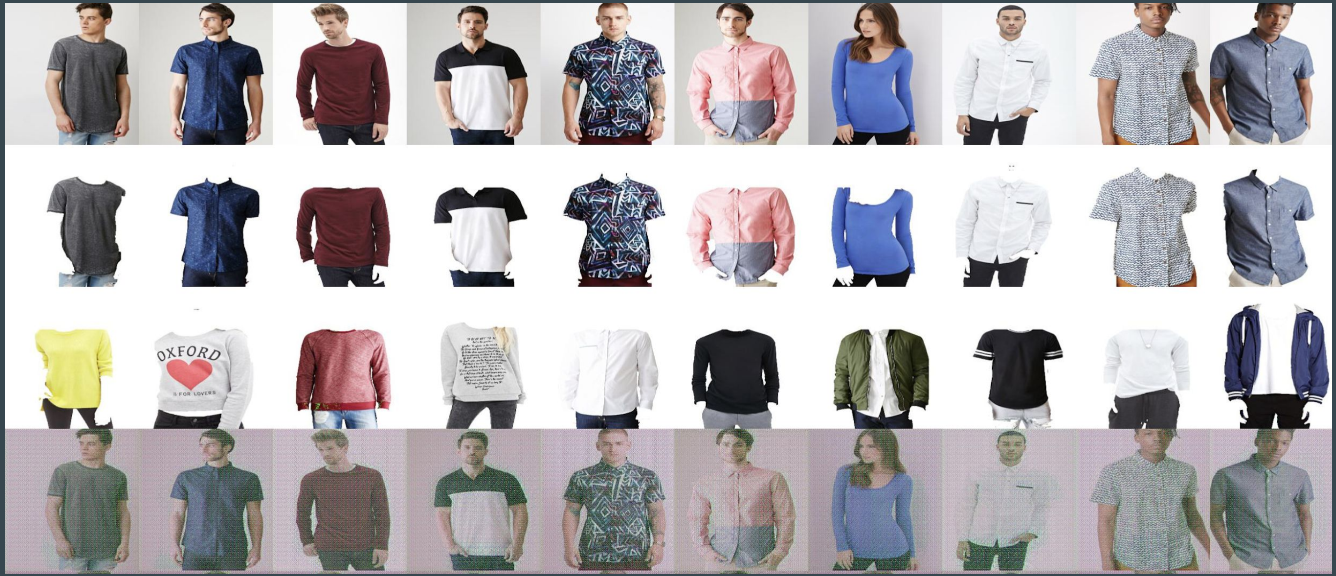
$$\mathcal{L}_{cyc}(G) = \mathbb{E}_{x_i, y_i, y_j \sim p_{\text{data}}} \left\| x_i - G(G(x_i, y_i, y_j), y_j, y_i) \right\|$$

x_i

G(G(x_i, y_i, y_j), y_j, _y_i)

Results (cross validation)

1-iteration



x_i

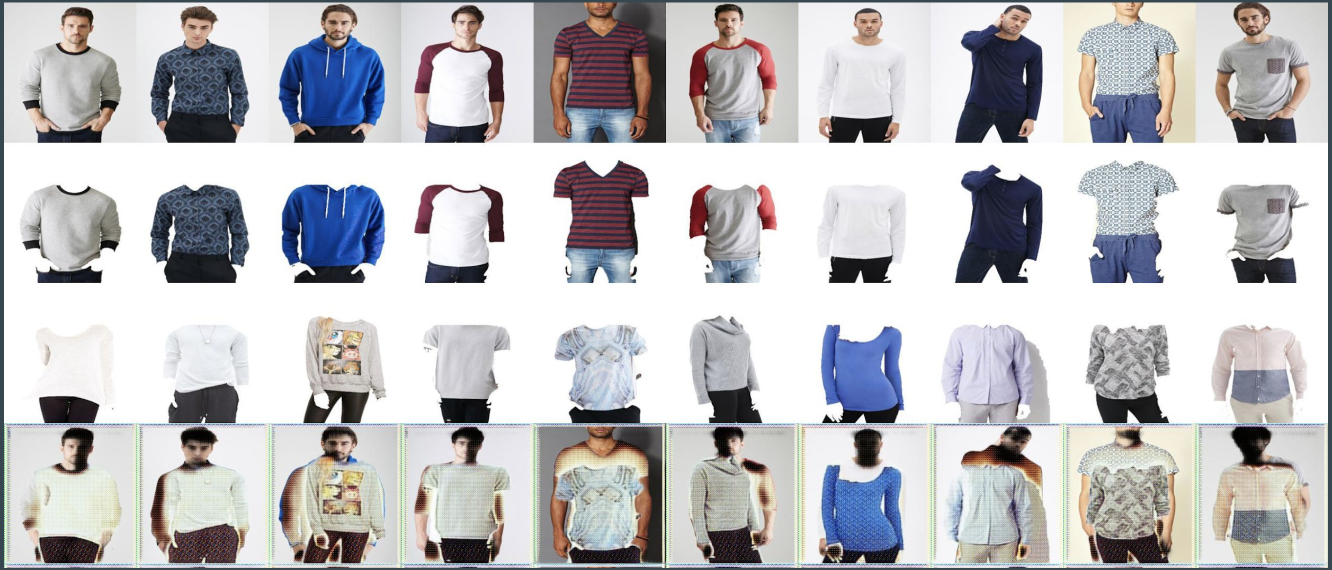
y_i

y_j

G(x_i, y_i, y_j)

Results (cross validation)

4000-iteration



x_i

y_i

y_j

G(x_i, y_i, y_j)

Results (cross validation)

9000-iteration

x_i

y_i

y_j

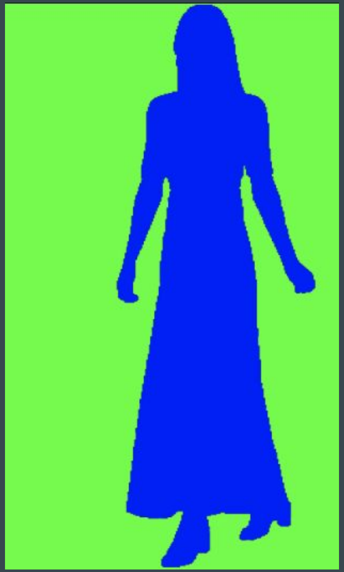
$G(x_i, y_i, y_j)$

Limitations

I. Computing Capability —— GPU & Money
II. Limited Training dataset (restricted by suitable pose & fashion & pose, fashion type)
III. Pose transfer cannot implement right now
IV. Girls are different and difficult in terms of fashion style and hairs
V. Positions are different...

Future work

We may consider centering the images



right side

center

THANK YOU