

5-29-2020

Virtual Try-On With Generative Adversarial Networks: A Taxonomical Survey

Andrew Jong
San Jose State University

Melody Moh
San Jose State University, melody.moh@sjsu.edu

Teng Sheng Moh
San Jose State University, teng.moh@sjsu.edu

Follow this and additional works at: https://scholarworks.sjsu.edu/faculty_rsca

Recommended Citation


Andrew Jong, Melody Moh, and Teng Sheng Moh. "Virtual Try-On With Generative Adversarial Networks: A Taxonomical Survey" *Advancements in Computer Vision Applications in Intelligent Systems and Multimedia Technologies* (2020): 76-100. <https://doi.org/10.4018/978-1-7998-4444-0.ch005>

This Contribution to a Book is brought to you for free and open access by SJSU ScholarWorks. It has been accepted for inclusion in Faculty Research, Scholarly, and Creative Activity by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.


Chapter 5

Virtual Try-On With Generative Adversarial Networks: A Taxonomical Survey


Andrew Jong

 <https://orcid.org/0000-0001-8457-8288>
San Jose State University, USA

Melody Moh

 <https://orcid.org/0000-0002-8313-6645>
San Jose State University, USA

Teng-Sheng Moh

 <https://orcid.org/0000-0002-2726-102X>
San Jose State University, USA

ABSTRACT

This chapter elaborates on using generative adversarial networks (GAN) for virtual try-on applications. It presents the first comprehensive survey on this topic. Virtual try-on represents a practical application of GANs and pixel translation, which improves on the techniques of virtual try-on prior to these new discoveries. This survey details the importance of virtual try-on systems and the history of virtual try-on; shows how GANs, pixel translation, and perceptual losses have influenced the field; and summarizes the latest research in creating virtual try-on systems. Additionally, the authors present the future directions of research to improve virtual try-on systems by making them usable, faster, more effective. By walking through the steps of virtual try-on from start to finish, the chapter aims to expose readers to key concepts shared by many GAN applications and to give readers a solid foundation to pursue further topics in GANs.

DOI: 10.4018/978-1-7998-4444-0.ch005

INTRODUCTION

In the past five years, Generative Adversarial Networks (GANs) have become a widely researched machine learning topic, especially for computer vision. *Google Scholar* returns over 12,000 results for “generative adversarial networks” when filtered between 2014, the year GANs gained traction, and 2019, the time of this chapter’s writing. This significant attention arose from research showing that GANs are capable of generating highly realistic output that closely approximates the original training distribution, to the point that the generated result may be indistinguishable to the human eye. One infamous example of this is *DeepFake*, the application that allows one to puppet a video of another person, to make it appear as if the other person were saying words contrary to what was originally said. In the era of Fake News, this is can be especially worrisome.

Yet GANs also have potentially beneficial applications. For instance, *DeepFake* has been used in the art installation *Dalí Lives* to reanimate the 20th-century painter Salvador Dalí for museum goers; other applications of GANs may help people visualize the future effects of climate change in their neighborhood and inform better understanding for the average citizen. The capability of GANs to generate high quality images on the fly has, consequently, opened new avenues of research in computer graphics as well as virtual, augmented, and mixed reality applications. From the perspective of artificial intelligence, the authors view GAN computer vision applications as most akin to visual imagination in humans, and thus are a worthy research pursuit.

Out of this vast range of applications, this chapter will focus on the application of GANs for Virtual Try-on, i.e. to allow users to virtually try-on digital clothing items at will. Though the chapter presents a narrow focus, virtual try-on shares core components with many GAN applications from photo editing to accelerated graphics rendering. By walking through the steps of virtual try-on from start to finish, the authors aim to expose the reader to key concepts shared by many GAN applications, and give the reader a solid foundation to pursue further topics in GANs in the future.

First, this chapter gives background to introduce virtual try-on and early approaches, but quickly moves on to explain GAN fundamentals, its core modifications, and other related work that set the stage for their application in virtual try-on. Next, the chapter details the evolution of select works from the first virtual try-on GAN to the most recent state-of-art that achieves try-on for video. This section shows the reader how the various components of complex GAN systems are managed together to achieve the feat of virtual try-on. Finally, the chapter concludes with an analysis of current issues and future research directions.

BACKGROUND

Motivation for Virtual Try-on

Worldwide, e-commerce accounts for one-third of clothing sales. Yet, simply because shoppers cannot tell how an article of clothing will look on them until it is tried on in person, e-commerce creates an immense source of waste. Nearly half of online shoppers return clothing due to unmet expectations, and every return adds a severe environmental impact from manufacturing, packaging, and transport. Even more drastically, companies often trash returns as a cheaper option than restocking items.

What if users could virtually try-on digital clothing before purchase? Beyond reducing the carbon footprint, Merle et al. (2012) suggests virtual try-on may improve the overall shopping experience by

boosting a shopper’s self esteem and positively influencing purchase intentions. The authors hope that mapping the progress in virtual try-on will stimulate further research towards achieving these goals.

Early Virtual Try-on History

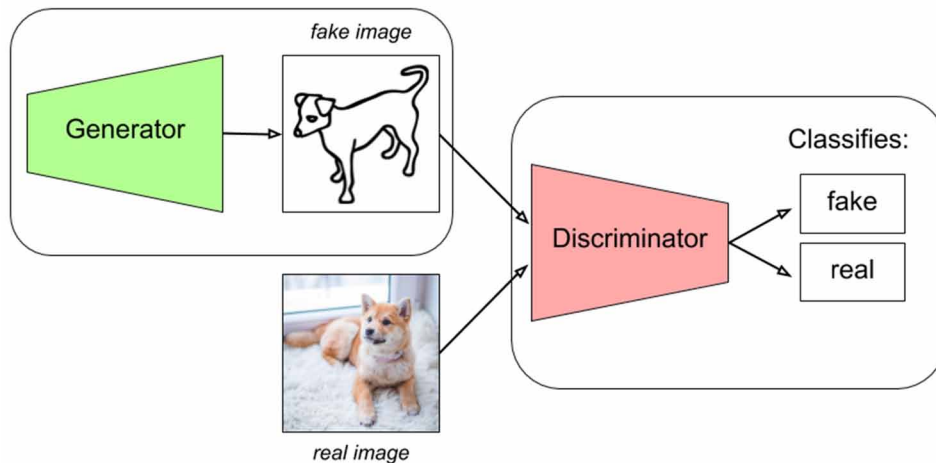
The earliest literature to use the term “virtual try-on” dates to 2001. Over the first decade of this century, virtual try-on focused on traditional computer graphics. The first virtual try-on methods either hand-modeled or 3D-scanned the user and clothes into 3D graphics for traditional rendering. While traditional 3D graphics may benefit from detailed cloth simulation, this technology was limited by the time it took to create 3D models.

However, one early work, Zeng et al. (2004), sought to algorithmically implement real-time try-on using only 2D images of a person and garment. Zeng’s approach uses a static (hand-designed and non-learnable) algorithm. The static algorithm circumvents the labor of 3D modeling or scanning by warping the garments to match the pose of the user. The user’s pose is detected using keypoints that inform the warping of the garments. The warped garment is then merged with the user’s image to produce the final result. Though Zeng’s approach deviated from traditional computer graphics methods at the time, it is the most similar to the modern approach used by GAN-based virtual try-on. The usage of keypoints and warping algorithm applied to 2D images is shared by today’s learnable GAN-based approaches.

Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a family of *generative* machine learning models, meaning they produce an observable X given a target class y : $P(X | Y=y)$. To take a computer vision example, suppose the target class y is “dog”, then the GAN would generate an image X to closely resemble a dog. GANs achieve this task using artificial neural networks. In particular, GANs use a system of two

Figure 1. This diagram of the Generative Adversarial Network shows its two-component system; the generator generates a fake image, while the discriminator distinguishes between real and fake



Virtual Try-On With Generative Adversarial Networks

neural networks: a generator (G) learns to produce fake images, in order to fool a discriminator (D) into believing that the fakes are real (Figure 1).

This two-component system gives rise to the ‘adversarial’ part of the name; in training, the error signal from the generator and discriminator inform the direction of the other’s optimization step, formally shown below. The Adversarial GAN loss was proposed by Goodfellow et al. (2014). The generator G aims to minimize the function, while the discriminator D aims to maximize it. These opposite goals form the adversarial nature of the GAN system. The distribution p_{data} is of the training data, while p_z is the generated distribution. Each network tries to outwit the other, leading to the production of high quality results.

$$\min_G \max_D V(D, X) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Conditional GANs come hand-in-hand with the original formulation to allow controlling the generator and discriminator with an input condition. The Conditional GAN equation was proposed by Mirza et al. (2014). Both domains are now conditional probabilities conditioned on y . For example, if generating images of dogs, the input condition could be a dog breed. Adjusting the breed condition would change the breed of the generated image.

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)}[\log D(x|y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|y)))]$$

On its own, the vanilla GAN, conditional or not, is insufficient to produce high quality results. Below, we detail important modifications that make GANs a feasible approach for virtual try-on. For further reading, a deeper explanation of GANs and their applications may be found in Pan et al. (2019). A detailed survey of notable, fundamental GAN advancements in computer vision may be found in Wang et al. (2019).

GANs for Pixel Translation (Pix2Pix)

A fundamentally important modification to the GAN was proposed by Isola et al. (2017). Image-to-Image translation, also known as ‘Pix2Pix’, is best described in the words of the original work:

In analogy to automatic language translation, we define automatic image-to-image translation as the task of translating one possible representation of a scene into another, given sufficient training data (Isola, 2014).

Pix2Pix builds upon the conditional GAN framework: it takes a conditional input image, such as a sketch, and translates it into another representation, such as a fully textured handbag. This framework lays the foundation for a vast scope of research applications in deep generative graphics, such as street maps to satellite images, black-and-white to color photos, 3D meshes to full renderings, and more. As it relates to this chapter, Pix2Pix is used in GAN-based virtual try-on, in which the input conditions are the user and garments, and the generated output is the user wearing the garment.

Two fundamentals of Pix2Pix are widely used in virtual try-on and other research applications. The fundamentals are: the addition of the L1 loss term and the usage of the U-Net architecture.

L1 Loss

The L1 loss essentially takes the absolute value difference between predicted and target values:

$$\langle_{L1} (G) = E_{x, y, z} [||y - G(x, z)||_1]$$

Recall the conditional GAN equation. Let this equation be represented with L_{cGAN} . Then the combined objective for the generator in Pix2Pix is:

$$G^* = \arg \min_G \max_D \varsigma_{cGAN}(G, D) + \lambda \varsigma_{L1}(G)$$

where λ is a hyperparameter to adjust the strength of the L1 contribution.

Adding the L1 loss to the generator forces the generated output to align with the overall structure and position of the ground truth target. If the output were not constrained, the task would deviate from translating to the true target. As an example with the handbags in Isola’s work, without L1 loss to maintain structure, the generated images may be incorrectly scaled, rotated, or moved, and not achieve the desired direct image-to-image translation. In virtual try-on, L1 loss is used to preserve the shape of the user’s pose in the generated image.

U-Net Architecture

The second fundamental component of Pix2Pix is the use of the U-Net architecture for the generator. U-Net, a fully convolutional neural network (F-CNN), was first introduced by Ronnenburg et al. (2015) for instance segmentation of cells in biomedical images.

Compared to previous bottleneck encoder-decoder F-CNNs, U-Net adds skip connections between corresponding encoder and decoder layers by concatenating their respective outputs. These skip connections function similarly to the skip connections from ResNet, a fundamental work from computer vision. As with ResNet, the presence of skip connections allows information from early layers to efficiently reach later layers. This also allows gradients from the loss function to backpropagate efficiently to avoid the vanishing gradient problem.

Furthermore, the larger convolutions of early encoder layers produce more information about low-level structure. Since low-level structure should be maintained in image-to-image translation, it is useful to pass this information forward to wield greater influence on the final result.

The Pix2Pix authors found that L1 loss and U-Net are essential to achieve image translation, and is the reason that many modern virtual try-on methods use these core components. The results of ablation studies are shown in Isola et al. 2017. The highest quality is achieved by combining all techniques.

Perceptual Loss Transfers Fine Detail

While Pix2Pix added the L1 loss to transfer low-level structures, this does not transfer high-level structures. The fine detail generated by Pix2Pix is entirely hallucinated by the GAN, not transferred from the input condition. However, realistic virtual try-on requires transferring of high-level structures such as cloth pattern, design, text, and texture. Perceptual losses are an important component to address this issue.

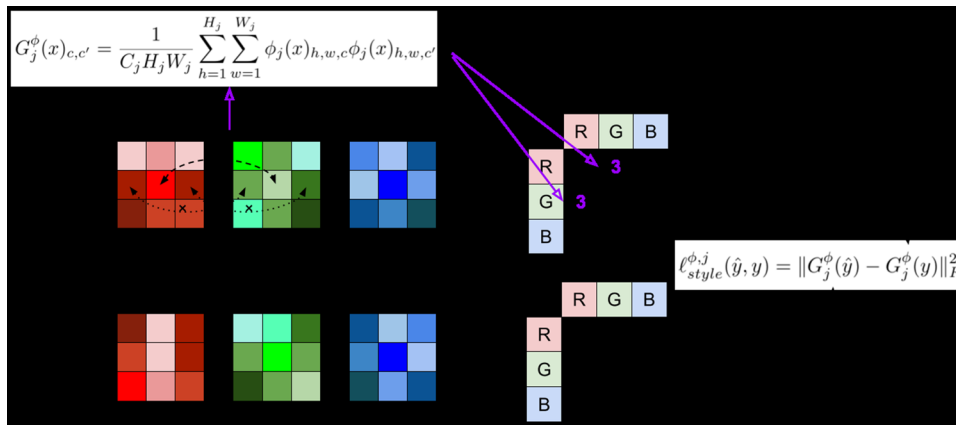
Gatys et al. (2016) introduced Style Loss to achieve detail preservation for style transfer, an application that transfers the style from paintings to photos (or vice versa). Johnson et al. (2016) further expanded on Style Loss by introducing Feature Loss, then categorized both as types of Perceptual Losses. The GAN virtual try-on literature often includes these perceptual losses in their generator’s objective. We detail both types below.

Style Loss

Style loss calculates a single scalar that measures the similarity of style between two images. For GANs, the two images are generated and ground truth images. The style of each image is captured by an $n_c \times n_c$ Gram matrix; then, a single scalar for the style loss is then calculated via the L2 distance between the matrices. See Figure 2 for a detailed example. The style loss function is:

$$\ell_{style}^{\phi,j}(\hat{y}, y) = \left\| G_j^\phi(\hat{y}) - G_j^\phi(y) \right\|_F^2$$

Figure 2. This diagram explains the calculation of style loss. The Gram ‘style’ matrices, $G(y)$ and $G(\hat{y})$, are calculated for the real and fake images respectively. Each cell value in the Gram matrix represents an inter-channel relationship within the image. The entire matrix represents the image style. In the example below, the Red-Green relationship is calculated; each dotted arrow represents one channel-wise product (for that coordinate). These products are summed to obtain the value 3 (purple). Due to the commutative property, the Gram matrix is symmetric ($G = G^T$). These two obtained matrices are compared using L2 loss to produce a single scalar that represents style loss.



$$G_j^\phi(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}$$

Feature Loss

Feature loss was used in Johnson et al. (2016) as another way to capture detail, specifically for the *content* of the image (for this reason, *feature loss* is sometimes called *content loss*). The intuition is that different convolutional layers activate to detect certain features in an image. Comparing these activations over a predetermined choice of layers with would measure the difference in content. Which layers to choose depends on the architecture and is a hyperparameter; in Johnson et al., the layers chosen from VGG19 were relu2_2, relu3_3, relu4_3, relu5_1, and relu5_3. Feature loss is formally defined as:

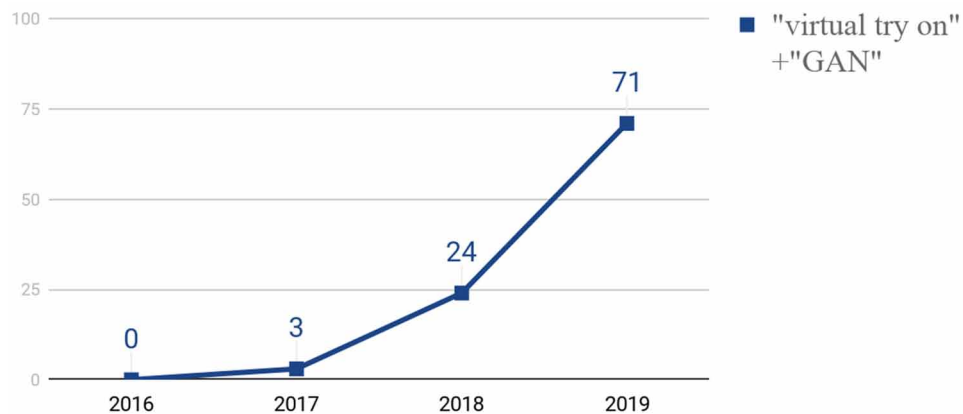
$$\ell_{feat}^{\phi,j}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \left\| \phi_j(\hat{y}) - \phi_j(y) \right\|_2^2$$

In practice, both style loss and feature loss may be added to the generator’s objective function with individually tunable weights.

VIRTUAL TRY-ON WITH GANS

With the astounding discoveries of GANs and pixel translation, the task of virtual try-on has increasingly been linked with GANs in recent years, as shown in Figure 3. Additionally, we note that there has been an academic movement to create fashion datasets of unprecedented size and quality. Human parsing tasks have shown to be more effective and can be used in conjunction to improve the quality of information that we can extract from the human body structure. Many models and networks, which are detailed below, have been created that have combined these works to create virtual try-on systems to transfer clothing.

Figure 3. Rise in results on Google Scholar for papers that include both “virtual try-on” and “GAN” together. Results were manually filtered by relevance and to remove duplicates.



Datasets

There has been a large proliferation of publically available datasets, such as the MVC, LIP, DeepFashion, and DeepFashion2 datasets, that have pushed forward the many component tasks of try-on mechanics. These datasets are of great importance to advance tasks, such as human parsing and pose estimation. Kuan-Hsien Liu, et al. (2016) created the MVC dataset, which contains more than 161K images of 37,499 clothing items. This dataset provides at least 4 different views for most clothing items in the dataset. This dataset also introduced a three-layer hierarchical attribute structure, which starts from gender, clothing category, and specific clothing attributes, such as color or pattern. Ke Gong, et al. (2017) introduced the LIP dataset, a group of 50,642 images which are annotated with 20 categories of human body parts and 16 different body joints. In addition, the LIP dataset contains images of humans from many different complex human poses, including occlusions, back-view, etc. Xiaodan Liang, et al. (2018) built upon the LIP dataset to include a combination of human parsing and pose annotations. These annotations led to more robust human parsing models, such as SSL and JPPNet.

Virtual Try-On With Generative Adversarial Networks

Ziwei Liu, et al. (2016) created the DeepFashion dataset contains 800K images collected from retail websites and Google. Each image is annotated from 50 clothing categories, with an additional 1,000 possible descriptive attributes. The images also have annotated landmarks, to effectively deal with deformation and pose variability of the clothing. The DeepFashion dataset also has 300K cross-pose or cross-domain image pairs, which adds to the robustness of the model. Yuying Ge, et al. (2019) created the DeepFashion2 dataset, which contains 801K images. 43.8K clothing identities are represented in 491K images, which averages out to almost 13 images per clothing type. DeepFashion2 is the only fashion database that has multiple pieces of clothing per image, and has up to 13 definitions of pose and landmarks. This allows there to be a large control of scale, occlusion, and perspective to create a very diverse dataset. The largest fashion database to date, DeepFashion2 supports the following tasks: clothes detection and classification, dense landmark and pose estimation, and cross-domain instance-level clothes retrieval.

Human Parsing

Clothing segmentation is a crucial task for virtual try-on. In order to execute a clothing from image A to image B, image A must undergo clothing segmentation, where each cloth in image A is identified. Then, one can impose each segment onto image B, based on the body segmentation of image B.

Cloth Segmentation

There have been quite a few proposed network architectures to accomplish cloth segmentation. The following describes the architecture details and loss functions of recent research on this topic.

FashionNet, proposed by Ziwei Liu, et al. (2016), is a deep model that takes up a network structure similar to that of VGG-16. The final layer of the VGG-16 model is replaced by three branches. The first branch captures the global features of the clothing image input. The second branch extracts local features over the estimated clothing landmarks. The third branch predicts these landmarks as well as visibility values. Every forward pass has three stages. In the first stage, the third branch is executed to predict landmarks in the input clothing image. Then, the second branch employs pool features to create local features that are invariant to deformations and occlusions in the input image. At the end of every forward pass, the outputs of the first and second branches are concatenated together in “fc7 fusion” to predict clothing category, attributes and model clothes pairs.

The backward pass defines four loss functions, for landmarks, visibility, category and attributes. After calculating these four loss values, the backward pass focuses on the blue branch as its main task. In doing so, it assigns higher weights to the landmark and visibility losses, while assigning lower weights to the attribute and category loss. It has been shown that this joint optimization leads to quicker convergence of the model. The second stage of the backward pass is to learn pairwise relations between clothing images as well as predict clothing categories and attributes.

Match R-CNN was proposed by Yuying Ge, et al. (2019). The network architecture has three main components: the Feature Network, Perception Network, and Matching Network. The Feature Network does feature extraction. ResNet-50 is a part of the Feature Network and it is responsible for the feature extraction, and the FPN uses a top-down architecture to create a pyramid of feature maps. RoI modules are responsible for extracting features from different levels of the pyramid feature map. The Perception Network has three goals: landmark estimation, clothes detection, and mask prediction. The landmark

estimation uses 8 convolution and 2 deconvolution layers to predict landmarks. The clothes detection uses one fully connected layer for classification and one fully connected layer for bounding box regression. The mask prediction uses 4 convolution layers, 1 deconvolution layer, and 1 convolution layer. Lastly, the Matching Network contains a feature extractor and a similarity learning network, which estimates the probability of two clothing items matching.

Pose Estimation and Body Segmentation

In analyzing existing approaches to human parsing, Ke Gong, et al. (2017) found that performance of human parsing at the time struggled with the back-view and occluded images. In contrast, parsing the upper-body is much more trivial because there are fewer semantic parts, which take up larger regions. Seeing how the head was a very important cue for human parsing tasks, Ke Gong, et al. (2017) and Xiaodan Liang, et al. (2018) proposed the self-supervised structure-sensitive learning (SSL) and JPPNet to create a more robust analysis of human images.

SSL incorporates joint structure identification into the human parsing loss. This allows the model to incorporate high-level information about human joint-structure, in tandem with the human parsing model. To enforce such knowledge, the joint structure loss weights the human parsing loss, to create a structure-sensitive loss. JPPNet is a joint human parsing a pose estimation network, which synthesizes local refinement and feature extraction connections to enable human parsing and pose estimation tasks in a way that is beneficial to each other. The parsing subnet, pose subnet and refinement network combine to unify the intrinsic connection between the two tasks.

Generating People in Clothing

Lassner et al. (2017) proposed to render clothed-humans with GANs from sampled from latent space, conditionable upon an arbitrary pose. Though not a virtual try-on technique per se, the concept to warp existing clothed humans to other poses is often used in person-to-person clothing try-on.

Lassner et al. makes two key contributions related to virtual try-on work. First is to introduce the concept of pose representation to conditionally generate cloth segmentations. Second is to use an image-to-image translation network to transform cloth segmentations into fully colored images.

The First GAN Try-on: Conditional Analogy GAN

To the authors' best knowledge, Jetchev et al. (2017) was the first to publish results for virtual clothing try-on using GANs. For this task, they aimed to exchange the existing clothes y_i , worn by a person x_i , with new clothes y_j . Supervised learning with a predetermined loss function would require thousands of examples of the same person wearing different clothes in the exact same pose; this dataset would be difficult to obtain as humans are not static objects and naturally move about. Thus, Jetchev et al. decided to use GANs for the discriminator's ability to learn a loss function and judge results in a self-supervised manner, as proven to work by Pix2Pix. They call their specific implementation the conditional analogy GAN (CAGAN).

The proposed method effectively paints over the clothes of the original person. Given a person x_i and a target of garment y_j , the generator produces both an image of the person wearing new clothes and an alpha mask. The alpha mask is then used to composite the original and generated images. Using a mask

Virtual Try-On With Generative Adversarial Networks

ensures the person’s original identity (such as facial features) is maintained, and that only the target garment is transferred without unduly affecting anything else.

The generator’s loss function for this training scheme is:

$$\min_G \max_D \varsigma_{cGAN}(G, D) + \gamma_i \varsigma_{id}(G) + \gamma_c \varsigma_{cyc}(G)$$

where L_{id} is a regularization term on the mask transparency, computed via the L1 norm of the mask (sum of absolute values). L_{id} is designed to limit the mask’s transparency and retain the original person’s features as closely as possible. However, this results in some undesirable blending of original and transferred textures. L_{cyc} is an additional cycle-consistency loss inspired by Zhu et al. (2017). Once the output of the person wearing new clothes is generated, that output is sent back to CAGAN to rewear the original clothes. The generated output should closely match the original image. The intuition of cycle consistency is to promote stability of the generated output.

Virtual Try-On Network (VITON)

Han et al. (2018) also approached GAN-based virtual try-on, becoming the second significant literature work to do so. They proposed three criteria:

- (1) body parts and pose of the person are the same as in the original image;
- (2) the clothing item in the product image deforms naturally, conditioned on the pose and body shape of the person;
- (3) detailed visual patterns of the desired product are clearly visible, which include not only low-level features like color and texture but also complicated graphics like embroidery, logo, etc. (Han, 2018).

CAGAN achieved criteria one with its alpha masking technique, but struggled to achieve criteria two and three. CAGAN’s weaknesses can be especially seen in comparisons of VITON and CAGAN results. Han et al. proposed VITON to address these issues. The authors of this chapter dedicate more attention to VITON as it became the baseline for many recent works.

Architecture

To address the clothing deformation criteria, VITON sought to explicitly warp the cloth, unlike CAGAN which had no dedicated process to do so. Thus VITON uses a two-stage system. The first stage is the Encoder-decoder Generator, which performs image-to-image translation with a person representation and target clothing as input conditions, and a coarse result and clothing mask as the output. The clothing mask is then used to warp the product image using a static shape-context matching algorithm (Belongie, 2002). The coarse result and warped cloth is then passed to a fully convolutional Refinement network to produce an alpha mask. The final image is obtained using the alpha mask to composite the warped cloth product and coarse person image, upon which a perceptual loss is calculated against the reference image.

Data Representation

A key difference proposed by VITON is to preprocess and include auxiliary information about the person's attributes, rather than to just use the original image. The person representation in VITON is composed of a pose map (detected using an existing keypoint detection network, body shape, and face-and-hair. The auxiliary information is then stacked along the channel dimension. This preprocessing offloads the work required by the image-to-image translation network to result in easier learning, but at the tradeoff of required computation for each input.

Perceptual Loss Function

Han et al. (2018) was the first to demonstrate that perceptual loss could be used in clothing try-on to encourage the transfer of texture detail. The loss function used was the feature loss (described in detail in Feature Loss subsection above). Comparatively, CAGAN relied only on a modified adversarial loss L_{cGAN} to generate detail. Including perceptual loss was instrumental to VITON's ability to transfer textured results, as can be seen in diagrams provided in the VITON work.

Results

A comparison diagram of VITON vs. CAGAN results may be found in Han et al. (2018), not shown here due to usage rights restrictions. Due to CAGAN's alpha masking, results for CAGAN often result in unrealistic blends of the original and target garment. CAGAN also sometimes struggles to place clothing on poses, instead misplacing the garments off-centered from the user. On the contrary, VITON addresses this issue with a static warping algorithm and refinement neural network. VITON results in consistent placement of the target garment on the user, as well as the transfer of patterns that span across the entire garment.

Characteristic Preserving Virtual Try-on

Wang et al. (2018) aimed to extend VITON by further improving the texture detail transfer, especially for complex and isolated pattern designs, logos, and text. While VITON transfers overall patterns, it struggles with detailed designs. To address this, Wang et al. propose a dedicated Geometric Matching Module, i.e. a neural network that learns to warp the clothes via supervised learning, rather than using the static shape-context algorithm used in VITON. The output of the Geometric Matching Module is then passed to a network similar to the Refinement network in VITON. The combined two-network system is named CP-VTON. Finding where to incorporate learning led to CP-VTON's higher quality results. For example, in VITON, logos and images would transfer blurred results. CP-VTON would better preserve these details and keep the text legible with the same visual characteristics such as color and shape. A visual comparison between these works can be found in Wang et al. (2018).

Multi-pose Guided Virtual Try-on

Dong et al. (2019a) sought to improve CP-VTON by improving large pose deformations, such as when the reference image is facing the opposite direction of the target clothes. These cases are tricky as the

Virtual Try-On With Generative Adversarial Networks

network must learn a sense of person orientation with only a 2D image. To achieve this, Dong et al. introduce a third stage, the Warp-GAN, that fits between the Geometric Matching Module and the Refinement network, and name the entire 3-stage system MG-VTON. This third stage learns to perform larger transformations than the Geometric Matching Module. When combined in this manner, MG-VTON learns to pose references better than prior work (see comparison figures in Dong et al. 2019a). MG-VTON is able to warp the reference image to the user’s pose while preserving the texture of the target clothes, even when the reference image pose is faced in the opposite direction.

SwapNet

Raj, A. et al. (2018) proposed the SwapNet, which is two-stage generative network that operates to transfer garment information from one image to another, while retaining clothing, body poses and shapes. The goal of this transfer is as follows:

Given an image A containing a person wearing desired clothing and an image B portraying another person in the target body shape and pose, we generate an image B composed of the same person as in B wearing the desired clothing in A. (Raj, 2018).

The first stage of the network method is the Warping module, which is similar to the Geometric Matching Module from CP-VTON. The warping module operates on the clothing segmentation of A and body segmentation of B to generate a segmentation of B that is consistent with the labels of B. This module goes through a dual path network. There is one encoder for the body segmentation, one encoder for the clothing segmentation and one decoder that combines the condensed representations. The loss values in the warping module as calculated below.

$$\varsigma_{CE} = -\sum_{c=1}^{18} 1(A_{cs}(i, j) = c) (\log(z_{cs}(i, j)))$$

$$\varsigma_{adv} = E_{x \sim p(A_{cs})} [D(x)] + E_{z \sim p(f1_{enc}(A_{cs}, B_{bs}))} [1 - D(f1_{dec}(z))]$$

$$\varsigma_{warp} = \varsigma_{CE} + \lambda_{adv} \varsigma_{adv}$$

The Texturing module takes on a U-Net architecture and the goal of this module is to obtain an embedding the desired clothing by ROI pooling on the 6 main body parts: torso, left/right arm, left/right leg, face. Then, this module generates and upsamples feature maps of original image, which are fed into the U-Net. This module uses clothing segmentation to control high-level structure and shape and clothing embedding to guide hallucination of low-level color and details. The loss values in the texturing module are calculated as below.

$$\begin{aligned}\varsigma_{L1} &= \|f2(z'_{cs}, A) - A\|_1 \\ \varsigma_{feat} &= \sum_l \lambda_l \|\phi_l(f2(z'_{cs}, A)) - \phi_l(A)\|_2 \\ \varsigma_{adv} &= E_{x \sim p(A)} [D(x)] + E_{z \sim p(f2_{enc}(z))} [1 - D(f2_{dec}(z))]\end{aligned}$$

SwapGAN

The SwapGAN, as proposed by Yu Liu, et al. (2019) proposed a multistage GAN model, consisting of three generators and one discriminator. The conditional image is the image with the model whose clothes we are trying to embed onto the person in the reference image. Four feature maps are extracted from the conditional image and reference image. Specifically, each image generates a pose feature map, a segmentation feature map, mask feature map, and head feature map. These feature maps are used as conditional inputs or as ground truth values to calculate loss values.

The first generator takes in an input of conditional image and the reference pose feature map and is used to manipulate the person in the conditional image to match pose and body shape of the reference image. This new generated image is the target image where the reference person wears the clothes in the conditional image, while preserving original pose and body shape. Mathematically, the equation for first generator is

$$X_{G1} = G_1(X_c, P_r)$$

and its corresponding loss is

$$\varsigma_{G1} = E_{X_c \sim p_{data}(X_c), P_r \sim p_{data}(P_r)} [(D(X_{G1}, X_c) - 1)^2]$$

The second generator is built on top of the first generator, and takes the output of the first generator as an input. The second generator also takes in the segmentation map of the conditional image as an input. This is used to make sure that the generated image is consistent with the original conditional image. The output of the second generator is a generated image that is consistent with the original conditional image. The main contribution of the second generator is that it considers the style of the clothes in the conditional image when transferring to the target image. Mathematically, the equation for first generator is

$$X_{G11} = G_{11}(X_{G1}, S_c) = G_{11}(G_1(X_c, P_r), S_c)$$

and its corresponding loss is

$$\varsigma_{G11} = E_{X_r \sim p_{data}(X_r), S_c \sim p_{data}(S_c)} \left[\left(D(X_r, X_{G11}) - 1 \right)^2 \right]$$

Virtual Try-On With Generative Adversarial Networks

The third generator is used to constrain the body shape of the generated images from the first and second generator. The loss is calculated by comparing the mask from the first and second generator with the mask feature map of the conditional image. The loss value is calculated through

$$\begin{aligned} \varsigma_{G111} = & E_{Mr-Pdata(Mr)} \left[\left\| M_{G111}(X_{G1}) - M_r \right\|_1 \right] \\ & + E_{Mc-Pdata(Mc)} \left[\left\| M_{G111}(X_{G11}) - M_c \right\|_1 \right] \end{aligned}$$

The total generation loss is calculated

$$\varsigma_G = \varsigma_{G1} + \varsigma_{G11} + \lambda \varsigma_{G111}$$

and the discriminator is trained to distinguish fake image pairs from real image pairs, as shown.

$$\begin{aligned} \varsigma_D = & E_{Xr-Pdata(X_r), Xc-Pdata(X_c)} \left[\left(D(X_r, X_c) - 1 \right)^2 \right] \\ & + E_{Xc-Pdata(X_c), Pr-Pdata(Pr)} \left[D(X_{G1}, X_c)^2 \right] \\ & + E_{Xr-Pdata(X_r), Sc-Pdata(Sc)} \left[D(X_r, X_{G11})^2 \right] \end{aligned}$$

An interesting note is that there is a post-processing step of embedding the head feature map from the reference image to the new generated image.

SwapGAN better handles pose deformations from the reference image. However, there has yet to be a published comparison between the MG-VTON and SwapGAN.

Fashion Model to Everyone: M2E-Try On Net

M2E-TON is a virtual try-on system proposed by Zhonghua Wu, et al. (2019). The M2E-TON has three sub-networks, the Pose Alignment Network (PAN), Texture Refinement Network (TRN), and Fitting Network (FTN). The goal of the PAN is to align the pose of the model image M with the target person P . The inputs into the PAN are the model image M , M_d , P_d , the dense pose estimations of the M and P , and M'_w , the model warped image. This conditional generative model generates M'_A , from M , with conditions on M , M_d , P_d , and M'_w . M_d and P_d are used to transfer poses from model M to target pose and the M'_w provides an effectively strong condition to generate clothing textures.

Before jumping into the TRN, we generate a merged image used the following equation:

$$\hat{M} = M'_w \odot R + M'_A \odot (1 - R)$$

The merged image will then be used in the TRN, which will help smooth the merged image, while preserving textual details of garments and produce a refined image M'_R . The FTN, then merges the

transferred image M'_R onto the person P , by obtaining regions of interest on person P to create a mask, based on dense pose estimation. This mask is applied on the texture-refine image, M'_R and P to generate the final image, P' . The M2E-TON uses something called unpaired-paired joint training. To ideally train these networks, we need pair images of M , P , P' . Due to the difficulty of acquiring this data, Zhonghua Wu, et al. (2019) proposes two training methods that need to be collaboratively used to train the network. First, the network should be trained on unpaired images, so that the discriminator learns to distinguish between real images and fake images. Then, use paired images of the same identity to train, to add structural correspondence between the input and output images.

Video Virtual Try-on

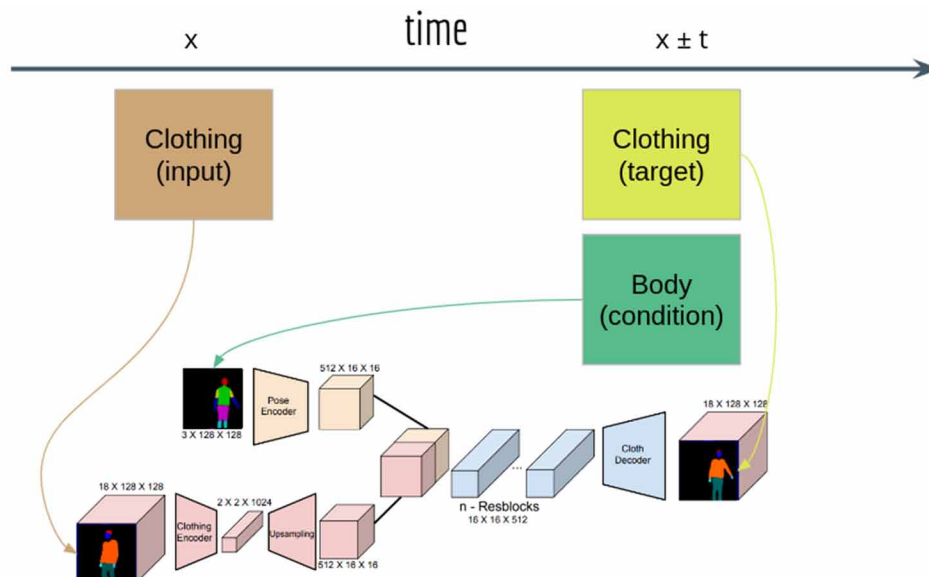
Given the significant development for virtual try-on with images, the next natural step is to investigate virtual try-on for video. Video try-on would allow a user to easily examine the clothing's appearance on their body at multiple angles, instead of needing to process individual images. However, video try-on adds new challenges, such as how to handle temporal consistency between video frames. By default, the image GAN has no relation between subsequent frames of a video, resulting in frame jumping and unrealistic distortions.

Video Frames for Augmentation

One work proposed by Jong et al. (2019) sought to investigate how the greater information provided in video data could augment the virtual try-on networks. Jong et al. expanded on the warp stage from

Figure 4. Jong et al. proposed that the cloth input condition be chosen at frame x , and a cloth target and corresponding body input condition be selected at frame $x + t$. This results in using the full video for learning cloth deformations.

Source: Jong, 2019



Virtual Try-On With Generative Adversarial Networks

Raj et al. (2018), and found that instead of relying on disjoint images of target outfits, using video to add slightly more data per clothing reference provides a stronger signal to learn cloth transformations (Figure 4), and significantly reduces the required amount of data overall for the warp stage. However, this preliminary work had yet to achieve full virtual try-on.

Complete Video Try-on

Dong et al. (2019b) was the first to achieve video clothing try-on for the entire try-on processing, including warp and texturing stages. The network is named Flow-Warping GAN (FW-GAN), for its incorporation of optical flow to address temporal consistency issues, and the warping of desired clothes and user to the video pose.

FW-GAN requires a video of a reference person wearing the desired clothes. This was collected in a new dataset, VVT, which was obtained from scraping fashion walk videos on fashion websites. The image of the user is then warped to match the pose of the reference, and the desired clothes from the reference are composited with the warped user.

The loss for the generator is as follows:

$$\mathcal{L}_{syn} = \alpha_1 \mathcal{L}_{gan} + \alpha_2 \mathcal{L}_{perceptual} + \alpha_3 \mathcal{L}_{pcl} + \alpha_4 \mathcal{L}_{flow} + \alpha_5 \mathcal{L}_{grid}$$

where \mathcal{L}_{gan} is the conditional adversarial loss, $\mathcal{L}_{perceptual}$ is the perceptual feature loss for texture transfer, \mathcal{L}_{pcl} is a parsing constraint loss that ensures the human part parsing at each generated frame is consistent, \mathcal{L}_{flow} is an optical flow loss that analyzes the last k frames to ensure temporal consistency, and \mathcal{L}_{grid} is a loss that functions similarly to the Geometric Matching Module from CP-VTON to ensure high quality cloth warping. Due to the video information present, FW-GAN is capable of synthesizing higher quality warping of clothes than previous still-image works such as VITON and CP-VTON.

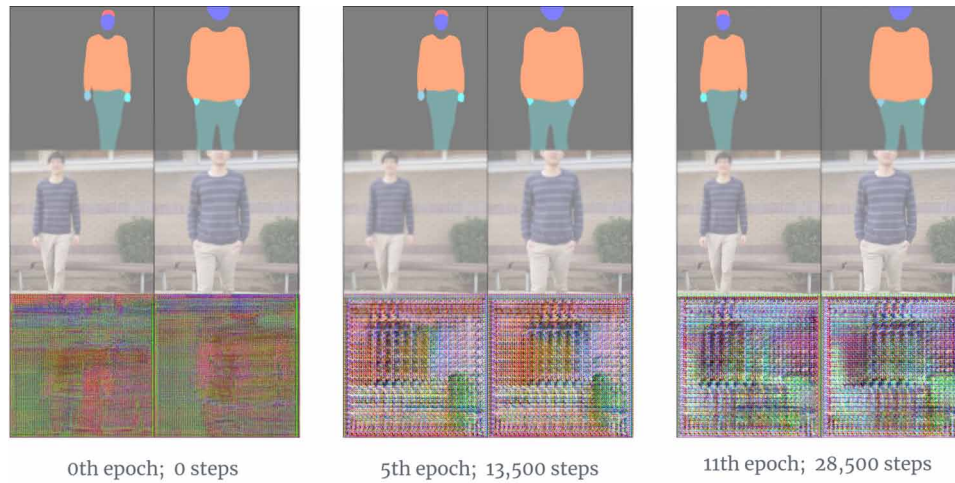
REPLICATION CASE STUDY

This section details the experiments of this chapter’s authors to replicate the texture stage from one of the virtual try-on works, SwapNet, described above. GANs, unlike other methods, can be especially tricky to achieve convergence and high quality results. Unlike in simpler tasks such as classification, the adversarial setup with the generator and discriminator produces a constantly changing loss landscape, making it difficult to achieve convergence. The purpose of this section is to show the reader how to employ critical thinking to conduct experiments for deep learning with generative adversarial networks. By reading through the process of these experiments, future readers can inform their own GAN training to overcome obstacles.

This section details the experiments to achieve cloth transfer results in the Texture Stage of SwapNet. The Texture Stage is the second neural network in the overall system. This stage aims to translate pre-warped clothing segmentations into textured images of the target clothes. After replicating the U-Net architecture of the texture stage, the authors first arrived at the results shown in Figure 5.

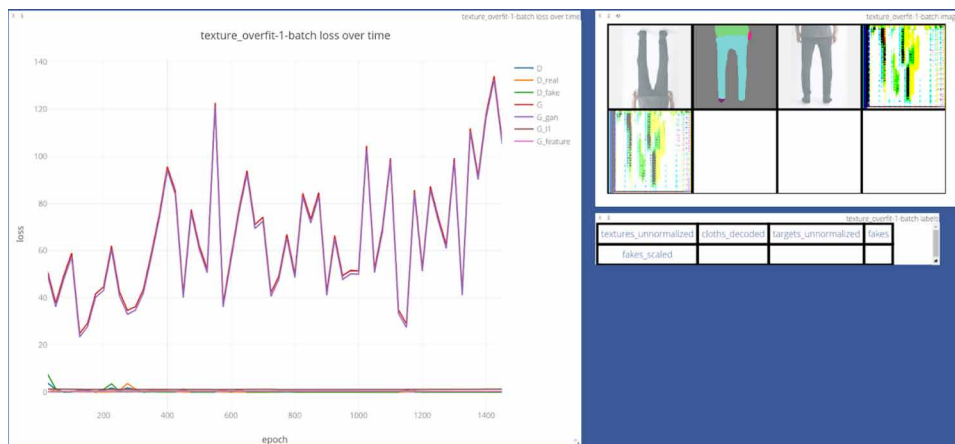
The first step to ascertain the root cause of failed convergence is to try and overfit one batch. A neural network should easily be able to overfit one batch of data; if it does not work, this implies something is

Figure 5. Initial results of texture stage in SwapNet replication. The top row shows the clothing segmentation. The middle row shows the target image to replicate the textures. The bottom row shows the generated output, which currently shows no discernible output.



wrong with the underlying pipeline. This is shown in Figure 6. As can be seen in the figure, even one batch did not converge properly, implying an error in underlying reimplement attempt.

Figure 6. Overfit 1 batch experiment. The left chart shows the generator loss diverging, while the discriminator loss has quickly settled at zero. While a zero-value loss is desirable in supervised training, having a zero loss in the discriminator for adversarial training disallows the generator from learning any further, resulting in divergence from the task.

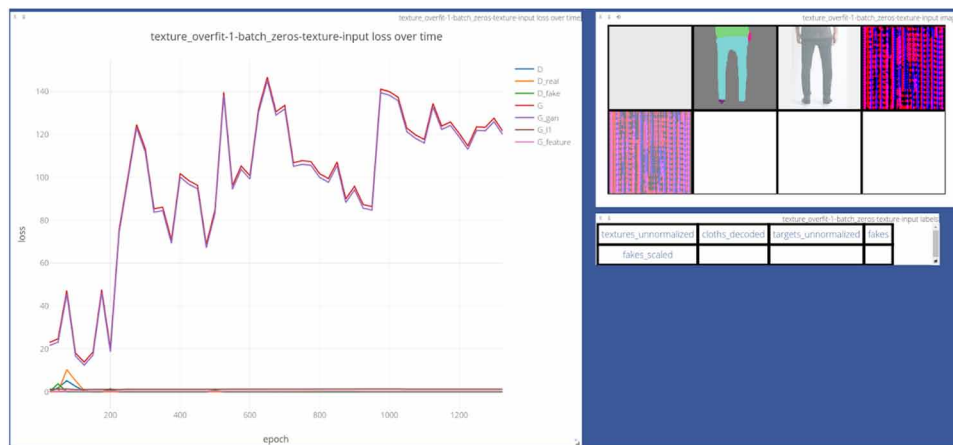


Because overfitting one batch did not work, the next step was to perform component testing of individual pipeline pieces. The input to the SwapNet’s texture stage generator consists of the (1) clothing segmentation, and (2) ROI pool values. To check that the issue was not caused by ROI pooling, the authors

Virtual Try-On With Generative Adversarial Networks

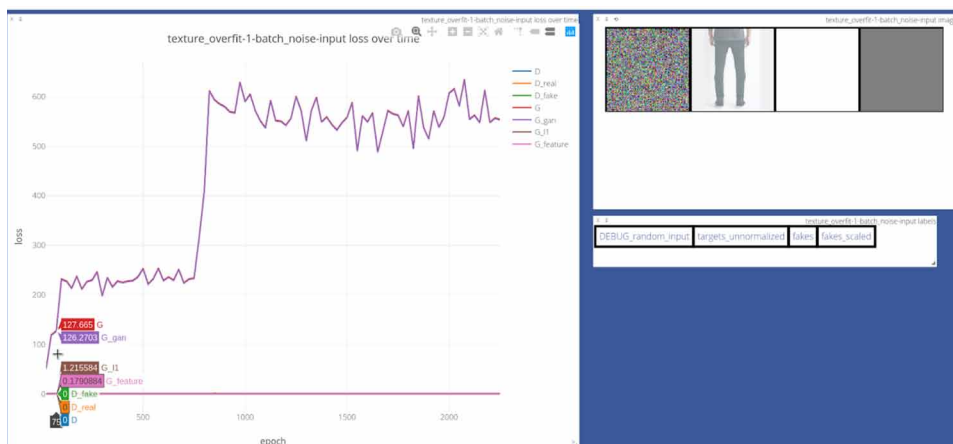
set the ROI pool input to zero-values. As can be seen in Figure 7, this still resulted in non-convergence and non-sensible output.

Figure 7. Results after ROI pool input is artificially set to zero



Recall that the texture stage is based on image-to-image translation from Isola et al. (2017), which uses a conditional generator to translate from one domain to another of a picture representation. To simplify the problem even further, the authors attempted to remove the conditional part of the GAN, and simply use a vanilla GAN setup to test convergence. The vanilla GAN takes as input noise and produces realistic images using the adversarial loss. While the images would not be conditioned on the clothes, observing this to work would inform experimental directions. As can be seen in Figure 8, this still resulted in no convergence. However, this was actually a flawed experiment, as setting the input to noise is not the only change needed to revert to a vanilla GAN. Recall that image-to-image translation uses auxiliary losses, such as the L1-loss, to encourage the output to converge towards desired target. This target is meant to

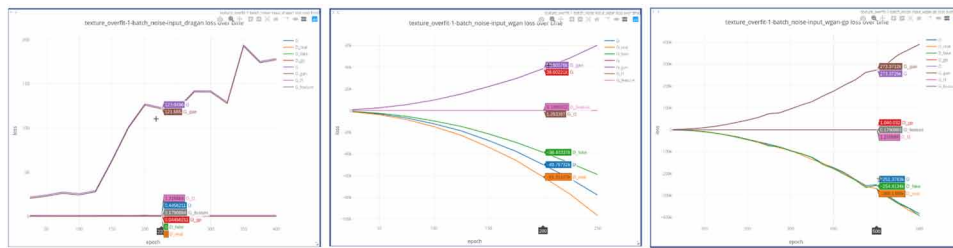
Figure 8. Attempt at vanilla GAN setup by setting input to only noise



be conditioned on the input. If the input is constantly changing, i.e. noise, then the network attempts to map random input to a particular output, without conditional context as to how to generate that output. This of course leads to no convergence. However, the authors did not realize this at the time.

After even the vanilla GAN setup did not work, the authors hypothesized that the problem could be rooted in the adversarial loss used in the vanilla GAN. Perhaps the dataset was too difficult for the vanilla GAN adversarial loss to converge on, which is known to have difficulties with convergence. This resulted in experiments with alternative adversarial loss functions, such as WGAN and DRAGAN. These experiments, shown in Figure 9, again resulted in no convergence.

Figure 9. Alternative loss functions did not converge either



Next, the authors attempted to experiment with hyperparameters of the auxiliary losses. In one experiment, the authors changed the L1 loss weight from 1 to 100. Suddenly, this resulted in discernible output with human body shapes, as can be seen in Figure 10. As another experiment, the authors set the

Figure 10. Experiment where L1 loss was set to 100



feature loss to zero. This again resulted in higher quality generated output as seen in Figure 11. Further results can be seen in Figure 12.

The results from the experiments imply that the underlying problem was in fact from a faulty implementation of the feature loss. When feature loss was set to 0, realistic shapes appeared quickly. However,

Virtual Try-On With Generative Adversarial Networks

as explained in earlier sections, feature loss is necessary for generating high quality detail. As can be seen in Figure 12, though rough shapes are achieved, details such as the face and garment details are missed. This is as far as the authors were able to achieve for the time being without public access to the code nor hyperparameters of SwapNet, delaying full reproduction of the work. This underscores the importance of releasing public code for reproducibility in deep learning.

Figure 11. Training with high L1 loss and no feature loss resulted in distinct body shapes

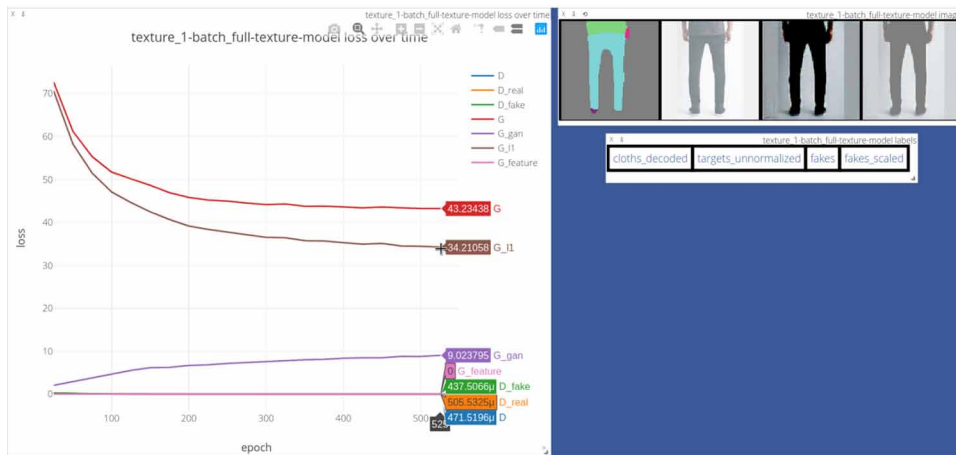
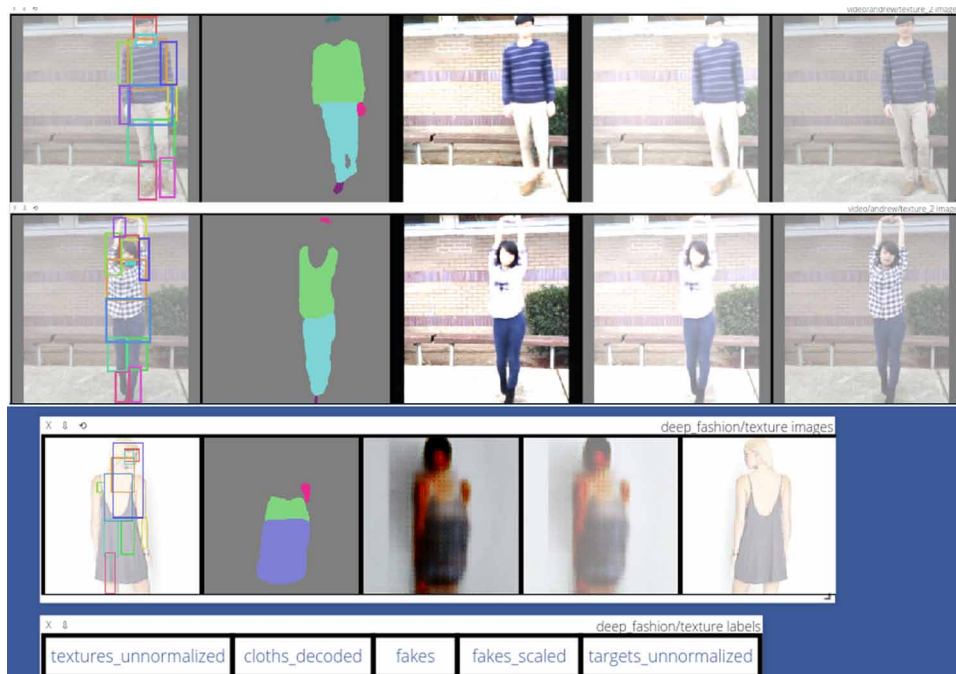


Figure 12. Results from training over a dozen epochs with high L1 loss and no feature loss.



FUTURE RESEARCH DIRECTIONS

Despite the growing number of works related to virtual try-on, there is still much work to be done that has yet to be addressed by existing work. We detail these directions below.

Usability

The usability of virtual try-on is under examined. Specifically, all existing research seeks to answer the question *Can it be done?*, such as *Can one generate high quality textures?*, *Can clothes be extracted directly from fashion models?*, and *Can one generate video try-on?*. While these fundamental questions are important, more investigation is needed into which directions promote practical usability. More questions such as *What type of quality do users focus on?*, *What gives a user the most information about clothing fit?*, and *How do these techniques generalize in the wild?* would be suitable for human-computer interaction research.

There need to be studies on how users will interact with such devices. Virtual try-on systems that use video input need to represent realistic cloth dynamics in the target video. There is also a lack of notion of how well the clothing in the model image fits the person in the target image. This will require more transfer between different-sized body types. We also wonder if it is possible for the target person to feel the material of the clothing in the model image.

There are many improvements to be made in improving resolution quality and effective transfer of clothes without facial or skin pigment distortion between model and target images. Additionally, the many approaches analyzed transfer the entire outfit from the model image, whereas a user would likely only wish to try on a particular clothing item, such as the shirt of the model image. Current research has made progress on improving transfer quality, even with high pose variance between the model and target images, but this quality needs to be further improved for practical application.

Inference Speed

The first virtual try-on GAN used a single end-to-end learning architecture that made inference quite fast. However, because the single architecture was weak quality, subsequent works have introduced preprocessing steps and multi-stage systems. Unfortunately, this prohibits any possibility of real-time try-on as every single input must be preprocessed, and separate networks must be loaded and offloaded from the GPU. If the field aims to achieve a real-time mirror interface, solutions must be found to bypass these drawbacks. Some possible solutions could be to use progressive growing and learning GAN architectures. Furthermore, research into more efficient architectures such as EfficientNet would have merit.

Adequate Comparison and Reproducibility

Despite the numerous papers released for virtual try-on, not all have released code to support their work. This limits the ability of research to compare existing methods effectively. Indeed, many works

encountered exclude comparison to a related literature simply because the code is not available. This is also a problem in machine learning as a whole that must be addressed.

Qualitative and quantitative comparison is also under developed. Current comparisons rely on the authors' opinion comparing images at low resolution (128x128), which is roughly one-tenth of high definition resolutions that people are accustomed to today. It is suspect how well texture quality can be compared at such low resolution. Quantitative metrics often involve crowd sourced user-preference studies, which are subject to anchor bias of the provided results, and do not offer much more evaluation than "one looks better than the other". More work is needed to develop metrics that evaluate specific types of quality, such as preservation of the user's identity, realism of cloth placement, detail, consistency across different users, etc.

CONCLUSION

Virtual try-on systems have existed since 2001. The early systems were heavily limited in their scalability and the intensive processing required for graphics. Since then, the discovery of GANs and pixel translation has shown to improve the quality of garment transfer. The chapter presented a focus virtual try-on by walking through the steps of from start to finish. The chapter explained key concepts shared by many GAN applications, and gave readers a solid foundation to pursue further topics in GANs. The chapter has successfully illustrated that, in combination with perceptual and style loss, building systems that are optimized to transfer clothes in high-quality style and perception has become possible.

Additionally, human-parsing tasks have become more robust, and have supplemented research towards more effective transfer techniques. Since then, there has been a large amount of research done on how to create different systems to learn how to warp the clothing from the model image to fit the body type in the target image. Then, there have been different intelligent networks developed to learn how to retain texture of the clothing that is warped onto the target image. These two modules appear in many different forms in the systems detailed above.

The chapter detailed the evolution of select works from the first virtual try-on GAN to the most recent state-of-the-arts that achieves try-on for videos. Virtual try-on systems are on the horizon, but there are many exciting problems that block these systems from being used everywhere. The quality of transfer needs to be improved, clothing transfer needs to give information about the fit of the item, improvements in speed of transfer to create real-time transfer are highly desired, and there needs to be more open-source code of released models and research.

REFERENCES

- Divivier, A., Trieb, R., Ebert, A., Hagen, H., Gross, C., Fuhrmann, A., & Luckas, V. (2004). *Virtual try-on topics in realistic, individualized dressing in virtual reality*. Academic Press.
- Dong, H., Liang, X., Shen, X., Wang, B., Lai, H., Zhu, J., ... Yin, J. (2019). Towards multi-pose guided virtual try-on network. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 9026-9035). IEEE.

- Dong, H., Liang, X., Shen, X., Wu, B., Chen, B. C., & Yin, J. (2019). FW-GAN: Flow-navigated Warping GAN for Video Virtual Try-on. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1161-1170). 10.1109/ICCV.2019.00125
- Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2414-2423). 10.1109/CVPR.2016.265
- Ge, Y., Zhang, R., Wang, X., Tang, X., & Luo, P. (2019). DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5337-5345). 10.1109/CVPR.2019.00548
- Gong, K., Liang, X., Zhang, D., Shen, X., & Lin, L. (2017). Look into Person: Self-Supervised Structure-Sensitive Learning and a New Benchmark for Human Parsing. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 10.1109/CVPR.2017.715
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680). Academic Press.
- Han, X., Wu, Z., Wu, Z., Yu, R., & Davis, L. S. (2018). Viton: An image-based virtual try-on network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7543-7552). IEEE.
- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134). IEEE.
- Jetchev, N., & Bergmann, U. (2017). The conditional analogy gan: Swapping fashion articles on people images. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2287-2292). 10.1109/ICCVW.2017.269
- Johnson, J., Alahi, A., & Fei-Fei, L. (2016, October). Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 694-711). Springer. 10.1007/978-3-319-46475-6_43
- Jong, A., & Moh, T. S. (2019). Short video datasets show potential for outfits in virtual reality. *Proceedings of the IEEE International Conference on High Performance Computing & Simulation (HPCS)*.
- Liang, X., Gong, K., Shen, X., & Lin, L. (2019). Look into Person: Joint Body Parsing & Pose Estimation Network and a New Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4), 871–885. doi:10.1109/TPAMI.2018.2820063 PMID:29994083
- Liu, K., Chen, T., & Chen, C. (2016). MVC: A Dataset for View-Invariant Clothing Retrieval and Attribute Prediction. *ICMR '16*.
- Liu, Y., Chen, W., Liu, L., & Lew, M. S. (2019). SwapGAN: A Multistage Generative Approach for Person-to-Person Fashion Style Transfer. *IEEE Transactions on Multimedia*, 21(9), 2209–2222. doi:10.1109/TMM.2019.2897897

Virtual Try-On With Generative Adversarial Networks

Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1096-1104). 10.1109/CVPR.2016.124

Mirza, M., & Osindero, S. (2014). *Conditional generative adversarial nets*. arXiv preprint arXiv:1411.1784

Raj, A., Sangkloy, P., Chang, H., Lu, J., Ceylan, D., & Hays, J. (2018). Swapnet: Garment transfer in single view images. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 666-682). Academic Press.

Wang, B., Zheng, H., Liang, X., Chen, Y., Lin, L., & Yang, M. (2018). Toward characteristic-preserving image-based virtual try-on network. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 589-604). Academic Press.

Wu, Z., Lin, G., Tao, Q., & Cai, J. (2019, October). M2e-try on net: Fashion from model to everyone. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 293-301). ACM. 10.1145/3343031.3351083

Zeng, X., Ding, Y., & Shao, S. (2009, December). Applying image warping technique to implement real-time virtual try-on based on person's 2D image. In *2009 Second International Symposium on Information Science and Engineering* (pp. 383-387). IEEE. 10.1109/ISISE.2009.9

ADDITIONAL READING

Merle, A., Senecal, S., & St-Onge, A. (2012). Whether and how virtual try-on influences consumer responses to an apparel web site. *International Journal of Electronic Commerce*, 16(3), 41–64. doi:10.2753/JEC1086-4415160302

Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., & Zheng, Y. (2019). Recent progress on generative adversarial networks (GANs): A survey. *IEEE Access: Practical Innovations, Open Solutions*, 7, 36322–36333. doi:10.1109/ACCESS.2019.2905015

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer. 10.1007/978-3-319-24574-4_28

Wang, Z., She, Q., & Ward, T. E. (2019). Generative adversarial networks: A survey and taxonomy. arXiv preprint arXiv:1906.01529.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232). 10.1109/ICCV.2017.244

KEY TERMS AND DEFINITIONS

Backpropagation: Neural networks use this technique to propagate the error signal to each of its neurons and evaluate the individual contribution of each neuron to that error.

Generative Adversarial Network: A family of generative models.

Human Parsing: A general task to parse understanding about humans, such to segment various parts of a human body. This includes Body Part Parsing (legs, torso, arms, etc.) and Clothing Part Parsing (hat, shirt, pants, etc.).

Loss Function: A function used to calculate the error of a neural network.

Neural Network: A class of machine learning technique vaguely inspired from biological neurons. Mostly synonymous with ‘deep learning’.

U-Net: A specific fully convolutional neural network architecture that employs skip connections between corresponding encoder and decoder layers after the bottleneck. Commonly used in generative adversarial network architectures.

Virtual Reality: A simulated experience that could either be realistic or different from the real world. Virtual reality differs from video games or movies in that virtual reality aims to be immersive.

Virtual Try-On: Using computers to allow users to virtually try-on digital clothing (or other items).

Warp Network: An auxiliary neural network used in virtual try-on GAN architectures that warp a cloth item to the user’s pose.