Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

FashionGAN: Display your fashion design using Conditional Generative Adversarial Nets

Y. R. Cui[1], Q. Liu[1], C. Y. Gao[†1] and Z. Su[1,2]

cuiyirui123m@gmail.com, liuq9967@163.com, mcsgcy@mail.sysu.edu.cn, suzhuo3@mail.sysu.edu.cn
[1] School of Data and Computer Science, Sun Yat-sen University, China
[2] National Engineering Research Center of Digital Life, Sun Yat-sen University

Figure 1: ***Some examples****. FashionGAN simplifies the pipeline of the traditional virtual garment display. It directly generates some specified garment images from the* ***garment design elements*** *like fashion sketches and fabric patterns. In addition, FashionGAN has the feature of reusability, which means the existing design elements (eg. contours or garment images) could be reused to generate another garment images.*

Abstract

Virtual garment display plays an important role in fashion design for it can directly show the design effect of the garment without having to make a sample garment like traditional clothing industry. In this paper, we propose an end-to-end virtual garment display method based on Conditional Generative Adversarial Networks. Different from existing 3D virtual garment methods which need complex interactions and domain-specific user knowledge, our method only need users to input a desired fashion sketch and a specified fabric image then the image of the virtual garment whose shape and texture are consistent with the input fashion sketch and fabric image can be shown out quickly and automatically. Moreover, it can also be extended to contour images and garment images, which further improves the reuse rate of fashion design. Compared with the existing image-to-image methods, the quality of images generated by our method is better in terms of color and shape.

CCS Concepts

•***Networks*** → *Network reliability;* •***Computing methodologies*** → ***Computer vision;***

1. Introduction

The workflow of garment production includes two main stages: design and manufacture. Before mass customized production, sample garments had to be produced to display the design concept. However, it is expensive and time-consuming to make a sample garment. In recent years, the personalized customizing of the garment becomes more and more fashionable, and requires the fashion enterprises to improve the workflow of garment production to meet the demand of garment customization. The pipeline of garment customization is the process of customer interactive selection of fabric and other garment design elements. Its core and key technology are to present the effect of the garment in the design phase. It will

† Corresponding author.

help the customer preview the virtual garment style through their chosen.

Plenty of display methods have been proposed to preview virtual garments in recent years. Most of them focus on displaying the 3D virtual garment. These methods display some details of the garment in the design stage from different views without making sample garments. They would reduce the cost of the research and development of the garment production. However, the pipeline of 3D virtual garment display contains a series of complex operations, such as making patterns from fashion sketches, 2D pattern sewing. Further, these operations need lots of experienced users' interactions.

Different from the 3D virtual garment display methods, we propose a novel end-to-end 2D virtual garment display framework with the conditional Generative Adversarial Nets. Users only need to input a fashion sketch and a fabric image into the framework. A garment image, whose shape is the same with the fashion sketch and texture follows the fabric pattern, will be automatically generated. It is very convenient for fashion designers and consumers to look through the realistic effect of a garment in the design phase. Thus, the proposed framework greatly improves the efficiency of the research and development of the garment production.

Generating a virtual garment image with a fashion sketch and a fabric image is a problem of **image-to-image translation**. Convolutional neural networks are powerful deep learning models that can solve such problems [ZIE16]. However, usual CNN architectures require pre-specified loss functions and such losses may not be suitable for some realistic images. In recent years, Generative Adversarial Network (GAN) [GPAM*14] and its improved models, conditional Generative Adversarial Network (cGAN) [MO14] has been widely used for image-to-image translation.

Pix2pix [IZZE17] first proposed uses cGAN with the input image condition to solve this problem. In pix2pix, shapes of the generated image follow the input, but its color and texture are random. In order to generate a deterministic result, BicycleGAN [ZZP*17] adds an encoder and establishes a bijection between the output space and the latent space. But the visual information contained in the latent vector still undetermined and inconsistent with the image condition of the generator. TextureGAN [XSA*18] tries to attach extra information (like cloth patch) to the image condition, but its results depend on the position and size of the attached patch. Above model trained by paired images. Recently, CycleGAN [ZPIE17] and MUNIT [HLBK18] propose using unsupervised learning to solve the image translation problem. A user obtains the desired image by specifying a content latent vector or transferring from the existing image. However, this method is hard to generate the image containing detailed texture. For example, a striped garment image in fashion design. We design a Generative Adversarial Network (GAN) architecture for previewing the garment design, named FashionGAN, which maps the desired fabric image to a selected fashion sketch or a contour image of the garment (see Figure 1). This network architecture is similar to BicycleGAN [ZZP*17], but we encode fabric image to generate latent vector and define a novel loss function which makes the network architecture more suitable for generating the virtual garment image.

Our main contributions are as follows:

- We propose an end-to-end 2D virtual garment display framework, which is seamlessly integrated with the existing methods of garment design and production in the fashion industry. The framework can automatically generate virtual garment images in the design phase without any complex operations and experienced user interactions, which can greatly improve the efficiency of garment design and production.
- By directly encoding the fabric image to generate latent codes containing visual information, our method could generate the garment image with a specified fabric image. While other existing image-to-image translation methods cannot establish the one-to-one map between the garment image and the fabric image. Furthermore, the encoding method of latent vector can ensure the generated garment image has similar fabric information even if the fabric image does not belong to the training dataset.
- Inspired by the architecture of BicycleGAN, we design a two-to-one GAN architecture which can generate an image with specified visual information, and we formulate a loss function to generator regular texture image, like stripe texture.

2. Related Work

Virtual garment display. Virtual garment display is useful in improving the efficiency of fashion design. The graphics researchers have investigated virtual garment display for decades, but most of their efforts emphasize dressed characters. These methods are mainly divided into two categories: 2D pattern-based garment modeling and Sketch-based garment modeling. 2D pattern-based garment modeling systems [VCMT05,FCRC05,BGK*13] allow users to edit patterns in 2D and use physics-based simulation to generate a 3D garment model. These systems require significant time and domain-specific user knowledge to create patterns that achieve a desired 3D garment. Sketch-based virtual garment modeling systems that allow users to trace garment silhouettes on top of a mannequin and then automatically infer plausible 3D garment shape [DJW*06,TWB*07,RMSC11,JHR*15]. These systems create garment models which are frequently simple, non-physical and hence unrealizable. Different from these 3D virtual garment display methods, we propose a 2D virtual garment display based on the generative adversarial model, which avoids the demand for domain-specific user knowledge and time-consuming of the traditional garment modeling.

Image-to-image GAN. Generative adversarial network (GAN) [GPAM*14] is an impressive model can generate an image from a noise vector. It is composed of the generator and discriminator which are trained in an adversarial manner. Due to the random output of the original GAN, researchers tried to add some conditions to constrain the generator. Conditional GAN (cGAN) [MO14] was one of those improved models. Types of conditions including discrete labels [MO14,DCSF15] , texts [RAY*16,ZXL*17] and so on. Especially, Isola et al. [IZZE17] first proposed regarding an image as the condition, which accomplishes image translation at the pixel level. Afterward, the cGAN with image condition has been successfully applied to many fields like style transfer [ZPIE17], terrain authoring [DPWB17], high-resolution image generating [WLZ*18], and so on. In order to generate the desired results, researchers

tried to add some stricter image conditions, such as a sketch with some colored scribblers [SLF*17], a sketch with masks and texture patches [XSA*18]. However, these methods were hard to generate uncommon shapes, color or texture [SLF*17]. Inspired by their ideas, we use a sketch to constrain the shape of the generated image. In addition, we also an encoded latent vector to constrain the generator output arbitrary color and texture. In order to generate diverse results, some researchers proposed multi-modal methods [ZZP*17, GKN*18, HLBK18]. Our FashionGAN is a cGAN architecture with the image condition as well. Instead of using ground truths and sketch/contour images as the paired image to train the network, we add a fabric pattern image to each pair so that the visual information of the fabric pattern will be mapped to contour image naturally. Except for cGAN, adding extra loss terms to the value function of GANs can achieve impressive image-to-image tasks like inpainting [PKD*16, YCYL*17] and super-resolution [LTH*17]. Rather than use the noise latent vector, VAE-GAN [LSLW16] adds a variational auto-encoder generating visual attribute vectors before the generator in original GAN.

3. Architecture of FashionGAN

To help users preview the effect of a garment in the design phase, the goal is to provide an image synthesis pipeline that can generate garment images based on a fashion sketch and a specified fabric image. Theoretically, this goal is to learn a mapping G among three different images domains $\mathcal{X},\mathcal{C},\mathcal{Y}$. During the training phase, three images in the three domains, $\boldsymbol{x}\in\mathcal{X},\boldsymbol{c}\in\mathcal{C},\boldsymbol{y}\in\mathcal{Y}$, are given. They are essentially samples of the joint distribution $p(\boldsymbol{x},\boldsymbol{c},\boldsymbol{y})$. During the test phase, let an image $\boldsymbol{x}\in\mathcal{X}$ and an image $\boldsymbol{c}\in\mathcal{C}$, then the generated image with the desired content is $G(\boldsymbol{x},\boldsymbol{c})\in\mathcal{Y}$. $G(\boldsymbol{x},\boldsymbol{c})$ is a sample of the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x}\boldsymbol{c})$. BicycleGAN provides a feasible way to establish the image synthesis pipeline. However, it is a one-to-one mapping between the latent vector and output space, which is not suitable for our goal completely. In the following, we will first review the BicycleGAN and further extend it to formulate our FashionGAN.

3.1. Preliminary

BicycleGAN learns a bijection between two image domains by considering cVAE-GAN [LSLW16] and cLR-GAN [DKD16] simultaneously. cVAE-GAN is actually a Variational Auto-Encoder (VAE), which reconstructs the ground truth $\boldsymbol{y}$ using the GAN loss $L_{GAN}(G,D,E)$ and the L1 loss $L_1^{im}(G,E)$. The encoded vector $E(\boldsymbol{y})$ is expected to follow the Gaussian distribution that makes the KL divergence item $L_{KL}(E)$ is needed. Thus $E:\boldsymbol{y}\rightarrow\boldsymbol{z},G:\{\boldsymbol{z},\boldsymbol{x}\}\rightarrow\hat{\boldsymbol{y}}$. cLR-GAN aims to ensure each random latent vector $\boldsymbol{z}$ can generate reasonable results using adversarial loss item $L_{GAN}(G,D)$ and L1 loss item $L_1^{vec}(G,E)$. Thus, $G:\boldsymbol{z}\rightarrow\hat{\boldsymbol{y}},E:\hat{\boldsymbol{y}}\rightarrow\hat{\boldsymbol{z}}$. cLR-GAN constructs a mapping between the output of G and the latent vector, but the discriminator could not see the ground truth $\boldsymbol{y}$. Hence, BicycleGAN takes advantage of cVAE-GAN and cLR-GAN:

$$\begin{aligned}G^*=\arg\min_G\max_D L_{GAN}(G,D,E)+\lambda_{im}L_1^{im}(G,E)+\lambda_{KL}L_{KL}(E)\\+L_{GAN}(G,D)+\lambda_{vec}L_1^{vec}(G,E).\end{aligned}\tag{1}$$

Enlightened by the idea of acquiring input information by encoding and carrying different features through a latent vector, we establish the mapping between the output and the random latent vector. The randomness of latent vectors ensures the diversity of output and encoding the output ensures the rationality of the latent vector.

BicycleGAN has two issues that make it impossible to be used directly in making our fashion design display framework. Firstly, in the cVAE-GAN of BicycleGAN, the latent vector of the input ground truth contains some shape information of the image. Meanwhile, the input contour image is the condition of the generator that also constrains the shape of the generated image. Shape information conveyed by two different inputs may cause redundancy and inconsistency. Furthermore, the ground truth is always unavailable, especially in fashion design fields. Secondly, texture details in fabric pattern cannot be restored with only L1 loss. Figure 5(d) demonstrates the G just generates the average color of the texture in the fabric pattern.

3.2. FashionGAN

In order to solve these two problems, we improve the model of BicycleGAN. We use the image of the real fabric pattern to train the encoder E so that only the material and color information of the image of real fabric pattern is contained in the latent vector. In other words, the shape of the final generated image is constrained by fashion sketch/contour image, and the color together with the material is constrained by the fabric pattern. Then we add an extra local loss module to control the generator synthesizing texture. For simplifying, we name our solution as FashionGAN.

FashionGAN learns a two-to-one mapping among three image domains. Namely, $E:\boldsymbol{c}\rightarrow\boldsymbol{z},G:\{\boldsymbol{z},\boldsymbol{x}\}\rightarrow\hat{\boldsymbol{y}}$ (see Figure 2). Based on BicycleGAN, we combine two types of networks called netA and netB. The input of netA and netB is a tuple including a contour image $\boldsymbol{x}$, a fabric pattern image $\boldsymbol{c}$ and corresponding ground truth (desired garment image) $\boldsymbol{y}$. $\boldsymbol{x}$ is extracted from $\boldsymbol{y}$ using Holistically-Nested Edge Detection (HED) [XT15]. $\boldsymbol{c}$ is randomly cropped from $\boldsymbol{y}$. NetA uses the cVAE-GAN architecture. At first, $\boldsymbol{c}$ is encoded by E so that the latent vector $\boldsymbol{z}$ contains the visual information of the fabric pattern. Then the cGAN procedures is conducted. Thus $E:\boldsymbol{c}\rightarrow\boldsymbol{z},G:\{\boldsymbol{z},\boldsymbol{x}\}\rightarrow\hat{\boldsymbol{y}}$. The sampling item in cGAN loss function should be modified to $\boldsymbol{z}\sim E(\boldsymbol{c})$:

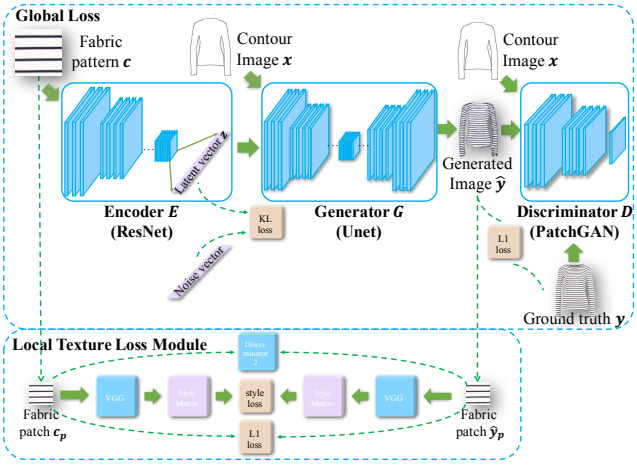
$$\begin{aligned}L_{GAN}(G,D,E)=&E_{\boldsymbol{x},\boldsymbol{y}\sim p_{data}(\boldsymbol{x},\boldsymbol{y})}[\log(D(\boldsymbol{x},\boldsymbol{y}))]\\&+E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x}),\boldsymbol{z}\sim E(\boldsymbol{c})}[\log(1-D(\boldsymbol{x},G(\boldsymbol{x},\boldsymbol{z})))].\end{aligned}\tag{2}$$

It is noted that, the original GAN loss may bring a mode collapse, which influences the diversity of results. In our training, we use WGAN [ACB17] loss, namely

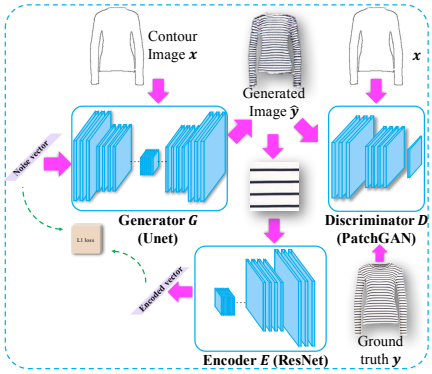
$$\begin{aligned}L_{GAN}(G,D,E)=&E_{\boldsymbol{x},\boldsymbol{y}\sim p_{data}(\boldsymbol{x},\boldsymbol{y})}[D(\boldsymbol{x},\boldsymbol{y})]\\&-E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x}),\boldsymbol{z}\sim E(\boldsymbol{c})}[D(\boldsymbol{x},G(\boldsymbol{x},\boldsymbol{z}))].\end{aligned}\tag{3}$$

We consider the L1 loss of image to ensure $\hat{\boldsymbol{y}}$ as close as $\boldsymbol{y}$, namely mapping the fabric pattern to the contour image. Sampling item is also modified:

$$L_1^{im}(G,E)=E_{\boldsymbol{x},\boldsymbol{y}\sim p_{data}(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{z}\sim E(\boldsymbol{c})}[\|\boldsymbol{y}-G(\boldsymbol{x},\boldsymbol{z})\|].\tag{4}$$



(a) Architecture of netA



(b) Architecture of netB

Figure 2: ***Architecture of netA and netB.*** *The generator, the discriminator and the encoder are same in the two architecture. The netA and netB are trained alternatively.*

We have demonstrated that BicycleGAN is hard to restore texture details in the fabric pattern using afore-motioned the global loss L_{GAN} and L_1^{im}. In order to describe the fine-grained texture, we add a local loss module referencing TextureGAN [XSA*18] to netA. These local loss items include local style loss L_s, local pixel loss L_p, and local discriminator loss L_{gan}. Different from TextureGAN, FashionGAN aims to encode the input fabric pattern then output the mapping result and diverse reference results. The architecture of FashionGAN is based on cVAE-GAN and cLR-GAN. Whereas, TextureGAN transform an input image attached with a fabric pattern to the realistic image. What they do is trying to "propagate" the texture patch in contour image. Its architecture is based on cGAN. Following the architecture of netA, G generates an image $\hat{\boldsymbol{y}}$ mapping the input fabric pattern $\boldsymbol{c}$. We sample two patches $\hat{\boldsymbol{y}}_{\boldsymbol{p}}$ and $\boldsymbol{c}_{\boldsymbol{p}}$ of random size from 45×45 to 65×65 at $\hat{\boldsymbol{y}}$ and $\boldsymbol{c}$.

The local style loss L_s, which is inspired by the deep learning based texture synthesis and style transfer work [GEB15, GEB16]. The style of an image can be represented by the Gram matrix of its feature maps, which is extracted from a pre-trained classification CNN. Namely, the inner product of vectorized feature maps $\boldsymbol{i}$ and $\boldsymbol{j}$ in layer the l:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l. \tag{5}$$

As CNN contains more than one convolution layer, we select different Gram matrices in multi-scale. Let G_{ij}^l and A_{ij}^l be the Gram matrix of $\hat{y}_p$ and c_p in layer i, respectively. Local scale loss is:

$$L_s = \frac{1}{|L|}\sum_{l\in L}\frac{1}{4N_l^2M_l^2}\sum_{i,j}(G_{ij}^l - A_{ij}^l)^2, \tag{6}$$

where N_l is the number of feature maps in layer i and M_l is the number of pixel in each feature map. L represents the chosen layers. Following with Gatys et al. [GEB15] , we choose some feature maps of $\hat{\boldsymbol{y}}$ and $\boldsymbol{c}$ on the layer conv2_2, conv3_2 and conv4_2 in the VGG-19 network. Using Equation (6) calculates the style loss among them.

The pixel loss. We use the L1 loss to ensure the generated texture same to it in the input fabric pattern. Namely,

$$L_p = \| \hat{\boldsymbol{y}}_{\boldsymbol{p}} - \boldsymbol{c}_{\boldsymbol{p}} \|_1 . \tag{7}$$

The local discriminator loss L_{gan} . We train another conditional discriminator D_2 judging whether two patches have the same textures. Same to D,G,E, the value function of D_2,G,E is:

$$\begin{aligned} L_{gan}(G,D_2,E) = {} & E_{\boldsymbol{c}_{\boldsymbol{p}}\sim p(\boldsymbol{c}_{\boldsymbol{p}})}[\log(D_2(\boldsymbol{c}_{\boldsymbol{p}},\boldsymbol{c}_{\boldsymbol{p}}))] \\ & + E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x}),\boldsymbol{c}_{\boldsymbol{p}}\sim p(\boldsymbol{c}_{\boldsymbol{p}}),\boldsymbol{z}\sim E(\boldsymbol{c})}[\log(1-D_2(\boldsymbol{c}_{\boldsymbol{p}},W[G(\boldsymbol{x},\boldsymbol{z})]))], \end{aligned} \tag{8}$$

where $G(\boldsymbol{x},\boldsymbol{z}) = \hat{\boldsymbol{y}}, W(\hat{\boldsymbol{y}}) = \hat{\boldsymbol{y}}_p$. A pair of patches$(\boldsymbol{c}_{\boldsymbol{p}},\boldsymbol{c}_{\boldsymbol{p}})$ that have the same textures is regarded as a positive example, otherwise $(\boldsymbol{c}_{\boldsymbol{p}},\hat{\boldsymbol{y}}_{\boldsymbol{p}})$ is regarded as a negative example.

To sum up, the local texture loss can consists of above three terms, namely:

$$L_t = \lambda_s L_s + \lambda_p L_p + \lambda_{gan} L_{gan}. \tag{9}$$

Therefore, the loss function of netA is:

$$L_{NetA}(G,D,E) = L_{GAN}(G,D,E) + \lambda_{im}L_1^{im}(G,E) + \lambda_{KL}L_{KL}(E) + L_t. \tag{10}$$

Same with BicycleGAN, the cLR-GAN is considered as our netB. In general, a fashion designer expects to see more reference results except for the desired one. NetB plays a role in increasing output diversity by inputting random noise, as well as ensuring each random latent vector $z \sim N(0,1)$ could generate a rational result. NetA maps a random noise to the generated image, namely $\boldsymbol{z}\to\hat{\boldsymbol{y}}$. NetB maps the generated image to the random noise. Namely $\hat{\boldsymbol{y}}\to\hat{\boldsymbol{z}}$. Multi-modal results can be obtained with netB. The value function of NetB is:

$$L_{NetB}(G,D,E) = L_{GAN}(G,D) + \lambda L_1^{vec}(G,E). \tag{11}$$

In summary, the overall value function of FashionGAN is:

$$L_{All} = L_{NetA}(G,D,E) + L_{NetB}(G,D,E) \tag{12}$$

Figure 2 shows the architecture of netA and netB. When training FashionGAN , $\{\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)},\boldsymbol{c}^{(i)}\}$ are inputed. G,D,E in netA

is trained once according to Equation (10) by executing forward and backward propagation. Then keeping the G, D, E and netB is trained once according to Equation (11). Repeating this alternation.

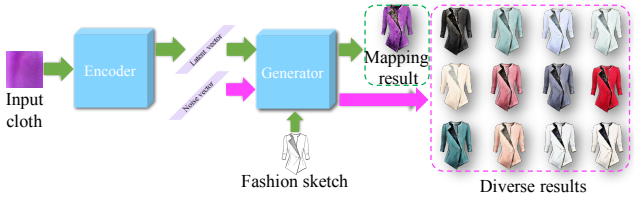


Figure 3: *Architecture of FashionGAN in the test phase*

Figure 3 shows the architecture of the FashionGAN in the test phase. Users just input a fashion sketch/contour of garment together with a fabric image. The generator will output a realistic garment image with the desired fabric. Users can also input a list of random latent vectors to show the diverse references.

4. Implementation

In this section, we first illustrate our garment dataset together with the experimental environment. Then we introduce the architecture and training details.

4.1. Dataset and experiment environment

We built a **garment dataset**. Garment images should have the clean background to avoid generating unreasonable results. Therefore we collected a lot of garment images and manually select the desired samples. Our data comes from two different sources:

- Crawling 5,000 images from the E-commerce website like taosousou [†].
- Filtering 19,000 images from public large-scale garment dataset like DeepFashion [‡] [LLQ*16].

Then we built a **fabric pattern dataset** by cropping a patch from each image in the garment dataset. In order to train G, D, E, garment image ground truths, corresponding contour images extracted by HED and fabric pattern images are inputted to FashionGAN . The training set contains 21000 pure color garments images and 1000 striped garment images. The test set contains 1800 pure color garment images and 100 striped garment images. We trained two different models on these two datasets. For each model, we trained 160 epochs on NVIDIA GeForce GTX 1080 Ti. At test time, we just input a fashion sketch/contour/garment image together with desired fabric pattern, the FashionGAN will output mapping results within 0.5s per image.

4.2. Architecture details

This subsection we illustrate the implementation details of encoder E, generator G and discriminator D. For E, we use ResNet [HZRS16] which is extensively employed in image classification and visual feature description. A 64×64 fabric pattern image input to E. For G, we also use U-Net architecture proposed in DCGAN [RMC15] which is the first CNN version of GAN. Inputs of G are a garment contour image with $256 \times 256 \times 1$ and an 8×1 encoded latent vector. When testing, a user can input arbitrary contour, fashion sketch or garment image to G. As for D. Following the pix2pix work, we utilize PatchGAN proposed by pix2pix for discriminator which can be understood as a form of texture/style loss. The network of G, D, E is composed of basic modules like residual or convolution blocks. It can be easily extended to higher resolution images. We also implement the 512×512 and 1024×1024 versions. In order to compare with baseline methods, we use 256×256 version.

4.3. Training details

In order to reduce mode collapse, we apply wGAN loss item [ACB17] to train G and D. According to the author's advice, we use RMSprop optimizer with 1e-4 learning rate and 1 batch size. For all our experiments, we use different weight settings for the losses. For netA, set $\lambda_{im} = 10, \lambda_{KL} = 0.01$ in Equation (10), $\lambda_s = 2,000, \lambda_p = 1, \lambda_gan = 1$ in Equation (9). For netB, we set $\lambda = 0.5$ in Equation (11).

5. Experiment results and analysis

This section, we give some baselines and evaluation metrics. We conduct comparative experiments on these baselines and give the qualitative and quantitative evaluation of the results. Lastly, We show other examples of fashion design, such as garment parts, handbags, shoes, and user-generated sketch.

5.1. Baselines

We compare five different the most related projects:

- **pix2pix** [IZZE17]. We used the paired data, ground truth-contour image, to train it.
- **CycleGAN** [ZPIE17]. It is an unsupervised model trained with adversarial loss and cycle reconstruction loss in two domains A and B. In our experiment, contour images are in domain A, and ground truths are in domain B.
- **BicycleGAN** [ZZP*17]. It can specify the content of the generated image by encoding ground truth. In addition, it can also generate multi-modal outputs by inputting random noise. We also use the paired data to train it.
- **MUNIT** [HLBK18]. It is also an unsupervised model which consists of two auto-encodersE_A, E_B and uses content and style latent code. In our experiment, contour image and ground truth input to E_A, E_B separately.
- **TextureGAN** [XSA*18]. It is a cGAN model whose condition is a contour image with a fabric pattern. We assume the fabric pattern is located in the center of the contour.

For the sake of fairness, we train all models 160 epoch.

† http://www.taotaosou.com/

‡ http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html

Figure 4: ***Example results of our method.*** *(a) Mapping a fabric pattern to a fashion sketch. (b) Mapping a fabric pattern to a garment image. (c) Mapping a fabric pattern to a contour image. (d) Mapping a stripe fabric pattern to a contour image. (e) Shoes, bags, garment parts and user-generated sketch mapping examples. The first column of each sub-figure represents the input contour, fashion sketch or garment image. The second column is the input fabric pattern. The third column is the corresponding mapping result. The rest columns contain more reference results.*

(a)

(b)

(c)

(d)

Figure 5: ***Results of our method and other methods.*** *(a) Mapping a fabric pattern to a fashion sketch. (b) Mapping a stripe fabric pattern to a contour image. (c) Reconstructing ground truth. (d) Mapping a fabric pattern to a contour image. Column 1-3 is our results. Follow columns are the results of pix2pix, BicycleGAN, MUNIT, CycleGAN. The last two columns show results of TextureGAN.*

5.2. Evaluate Metric

We give quantitative analysis metric to evaluate our model and the baselines.

Human preference. To compare mapping results by different methods. We perform the human perceptual study on the website of our laboratory. 30 volunteers are given 100 random input image pairs including contour/fashion sketch and corresponding fabric patterns, meanwhile, they are given 6 different output results for each pair. Then they grade 6 outputs in color accuracy, shape accuracy, and overall realism. The highest score is 5 and the lowest is 0. Lastly, we average these scores.

Inception score. The inception score (IS) [SVI*16, SGZ*16] is a well-known metric for evaluating generate results quantitatively from labelled dataset:

$$\exp(E_x KL(p(y|x) \parallel p^*(y))) \tag{13}$$

where x is one sample, and $p(y|x)$ is the softmax layer output of a trained classifier. $p^*(y)$ is the overall label distribution of generated samples. A higher inception score means high-quality and diverse.

5.3. Comparison against baselines

In this subsection, we first conduct 4 different groups of experiments (Figure 4) to illustrate our model. Then we conduct comparative experiments on baselines and give qualitative and quantitative analysis. Lastly, we present more general results of our model.

Figure 4 (a), (b), (c) presents some examples of mapping a fabric pattern to a fashion sketch, garment image or contour. And Figure 4 (d) show examples of mapping a stripe fabric pattern to a contour. Results demonstrate that FashionGAN can generate desired, realistic and reasonable garment images. In particular, clear body curves can be observed in the third rows of Figure 4 (c), therefore FashionGAN can also recognize men's and women's garments. Except for single color fabric pattern mapping problem, FashionGAN can also map stripe pattern since it contains local loss module.

Figure 5 compares FashionGAN with pix2pix [IZZE17], BicycleGAN [ZZP*17], MUNIT [HLBK18], CycleGAN [ZPIE17] and TextureGAN [XSA*18]. We retrain these models on our dataset containing 24,000+ garment images.

Figure 5 (a) (d) shows the fashion sketch and the contour image mapping examples. Except for pix2pix, all mapping results keep the outline of input fashion sketch. However, only our model mapping the color correctly meanwhile keeps the internal details. Pix2pix can generate realistic images, but the color is totally mismatched to input fabric pattern. The reason is that pix2pix try to build a one-to-one mapping between the ground truth and the contour image. When the ground truths do not exist, they cannot generate the desired output. BicycleGAN generates the mapping result that keeps the color partially for the encoded latent vector contains part of visual information of the fabric pattern. Results of MUNIT can keep the color, but the shape especially internal details of generated images are deficient. MUNIT model is trained in an unsupervised manner, and using style code to control the results. The style encoder of MUNIT includes several strode convolutional layers, followed by a global average pooling layer and a fully connected (FC) layer [HLBK18]. This architecture is good for natural image, but is not suitable for the fashion sketch or contour image. Results of CycleGAN are the worst. It cannot map the color correctly. TextureGAN tries to "propagate" the input fabric pattern. The results depend on the size and position of the pattern. When the pattern is larger and at the center of fashion sketch, the filling results will be better. In addition, Xian et al. [XSA*18] state that they need to specify the image mask of the filling region, which brings complex operations to the user. Compared with these methods, our FashionGAN does not need the mask. The color and texture of the generated image are controlled by the encoded latent vector, and the shape constrained by fashion sketch. Figure 5 (b) shows the stripe fashion sketch mapping examples. We can observe that pix2pix cannot correctly map shape and color. CycleGAN cannot inherit the color of the fabric pattern. BicycleGAN cannot map the detail of strip fabric pattern into contour. Its results are close to the average color of the input pattern. Results of MUNIT keep the stripe but not smooth enough compared with ours. The reasons for these experimental results same to (a). To a certain extent, the texture can be propagated using TextureGAN. For regular texture like the stripe, however, this level of filling will influence the visual effect. Figure 5 (c) shows some examples of reconstructing the ground truth. We found that our model and CycleGAN can reconstruct the shape and color of ground truth well. See the second row of Figure 5 (c), the result of pix2pix and BicycleGAN exists the color deficiency. The result of MUNIT and TextureGAN is relatively worse.

Table 1: ***Human performance***

	Fashion-GAN	pix2pix	Bicycle-GAN	Cycle-GAN	MUNIT	Texture-GAN
Color accuracy	4.573	1.898	3.395	1.778	3.193	3.223
Shape accuracy	4.224	3.332	4.167	2.716	3.024	3.194
Overall feeling	4.381	2.548	3.739	2.083	2.441	3.138

Table 1 lists the average score of color, shape accuracy and overall feeling to the generated garment images. First, the unsupervised model (CycleGAN, MUNIT) got a lower score, indicating that they are not appropriate for specified image-to-image translation. Then, the supervised model achieves a higher score in shape accuracy. We illustrated that they both belong to cGAN. The input contour is the constraint condition of the generator causing the generated image keeps the garment shapes. Lastly, FashionGAN , BicycleGAN, MUNIT, TextureGAN got higher color accuracy, indicating that if the color information provided, the generator can output colorful garment images. MUNIT and TextureGAN try to convey this information through conditions in the generator. FashionGAN and BicycleGAN use the latent vector conveys this information. BicycleGAN encodes the input ground truth and its generator is constrained by the contour image. There always exist redundancy and inconsistency such as contour information in latent vector and condition. Therefore, we utilize encoder express visual information containing fabric pattern, and using the condition of the generator to convey shape information. Experimental results proved that it is the most effective way.

Table 2: ***Inception score***

	Fashion-GAN	pix2pix	Bicycle-GAN	Cycle-GAN	MUNIT	Texture-GAN
Contour+ cloth patch	3.408	3.454	3.368	3.680	3.466	3.171
Contour+ ground truth	2.643	2.071	2.396	2.191	2.498	2.282
Fashion sketch+ cloth patch	4.567	4.393	4.502	4.184	5.074	3.840
Contour+ stripe patch	3.097	3.183	3.421	3.128	3.323	3.383
Average	3.429	3.275	3.422	3.296	3.59	3.169

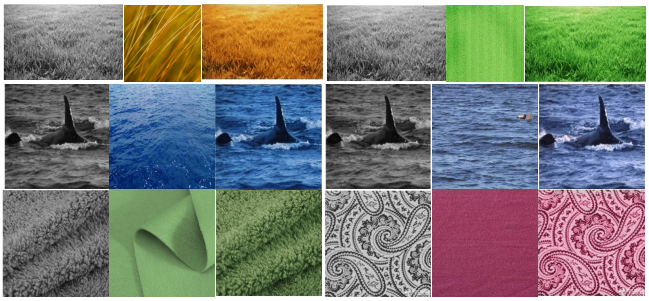


Figure 6: *Examples of general image-to-image results.*

Figure 7: *Examples of some failed mapping results.*

Table 2 displays the inception score of different methods in 4 test cases. This score reflects the diversity and quality of the generated images. Except for MUNIT, our FashionGAN is better than other models. TextureGAN and pix2pix are one-to-one mapping models, thus their inception scores are lower. Our multi-modal module reference the BicycleGAN, and their inception scores are very close. MUNIT is the state-of-the-art multi-modal model, which achieves the highest IS. The reason is that they assume all images share content space. In the future work, we plan to add a content encoder to the diversity of results.

5.4. More applications

FashionGAN is a robust model. Figure 4 (e) shows some mapping examples of garment parts and user-generated sketches. These test cases are totally different with training data, but they are still mapped correctly with the model trained on garment dataset. Except for garment design, shoes and bags design are very common in fashion fields. We also retrain our FashionGAN on shoes and bags dataset provided by BicycleGAN [ZZP*17]. Figure 4 gives some examples of test cases. FashionGAN can generate desired results by specifying the fabric pattern. FashionGAN can be extended to general image-to-image translation like colorization. Figure 6 shows examples like changing the color of seawater, grass or a fabric pattern.

5.5. Limitation and future work

Figure 7 shows some failed mapping examples. Our FashionGAN is workable for single-color and regular (like stripe) fabric pattern, but it cannot map an irregular one. Especially, we selected 2,000 garment images with an irregular texture and trained them on FashionGAN. The generator outputs images with the average color of the input fabric pattern. We analyze the reasons are as follows. First, the encoder in our network can learn the mapping between single-color or simple color combination (like stripe) and latent vector. There are various kinds of irregular textures in the real scene. Each of them contains complex color combinations. It is hard to describe it using latent vector in the current architecture. Second, the irregular textures are in variant shapes, the current local loss module cannot reconstruct them well. In the future, we aim to construct a new local loss module describing the irregular texture. What' more, we will extend our model to more general image translation tasks, like image colorization and style transfer.

6. Conclusion

We propose FashionGAN for exhibiting the design effect of garments. Our model is an end-to-end framework which only needs users input a desired fabric image and a fashion sketch, a garment image based on input data will be automatically generated. Based on cGAN, FashionGAN establishes a bijection relationship between fabric and latent vector so that the latent vector can be explained with fabric information. Moreover, we explore to add some local losses to help generate images with stripe texture. The experiments show our method can indeed achieve impressive results in fashion design.

7. Acknowledgements

This research is supported by the National Key Research and Development Plan in China (2018YFC0830500), the Guangzhou Science Technology and Innovation Commission (GZSTI16EG14/201704030079), and the National Natural Science Foundation of China (61772140).

References

[ACB17] ARJOVSKY M., CHINTALA S., BOTTOU L.: Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017). 3, 5

[BGK*13] BERTHOUZOZ F., GARG A., KAUFMAN D. M., GRINSPUN E., AGRAWALA M.: Parsing sewing patterns into 3d garments. *Acm Transactions on Graphics(TOG) 32*, 4 (2013), 1–12. 2

[DCSF15] DENTON E., CHINTALA S., SZLAM A., FERGUS R.: Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems(NIPS)* (2015), pp. 1486–1494. 2

[DJW*06] DECAUDIN P., JULIUS D., WITHER J., BOISSIEUX L., SHEFFER A., CANI M.-P.: Virtual garments: A fully geometric approach for clothing design. In *Computer Graphics Forum* (2006), vol. 25, Wiley Online Library, pp. 625–634. 2

[DKD16] DONAHUE J., KRÄHENBÜHL P., DARRELL T.: Adversarial feature learning. *arXiv preprint arXiv:1605.09782* (2016). 3

[DPWB17] DIGNE J., PEYTAVIE A., WOLF C., BENES B.: Interactive example-based terrain authoring with conditional generative adversarial networks. *Acm Transactions on Graphics(TOG) 36*, 6 (2017), 228. 2

[FCRC05] FONTANA M., CARUBELLI A., RIZZI C., CUGINI U.: Clothassembler: a cad module for feature-based garment pattern assembly. *Computer-Aided Design and Applications 2*, 6 (2005), 795–804. 2

[GEB15] GATYS L. A., ECKER A. S., BETHGE M.: Texture synthesis using convolutional neural networks. pp. 262–270. 4

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition(CVPR)* (2016), pp. 2414–2423. 4

[GKN*18] GHOSH A., KULHARIA V., NAMBOODIRI V. P., TORR P. H., DOKANIA P. K.: Multi-agent diverse generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 3

[GPAM*14] GOODFELLOW I. J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in Neural Information Processing Systems(NIPS)* (2014), pp. 2672–2680. 2

[HLBK18] HUANG X., LIU M.-Y., BELONGIE S., KAUTZ J.: Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision(ECCV)* (2018). 2, 3, 5, 9

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). 5

[IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 2, 5, 9

[JHR*15] JUNG A., HAHMANN S., ROHMER D., BEGAULT A., BOISSIEUX L., CANI M. P.: Sketching folds:developable surfaces from non-planar silhouettes. *Acm Transactions on Graphics(TOG) 34*, 5 (2015), 1–12. 2

[LLQ*16] LIU Z., LUO P., QIU S., WANG X., TANG X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). 5

[LSLW16] LARSEN A. B. L., SONDERBY S. K., LAROCHELLE H., WINTHER O.: Autoencoding beyond pixels using a learned similarity metric. *International Conference on Machine Learning(ICML)* (2016), 1558–1566. 3

[LTH*17] LEDIG C., THEIS L., HUSZAR F., CABALLERO J., CUNNINGHAM A., ACOSTA A., AITKEN A., TEJANI A., TOTZ J., WANG Z., SHI W.: Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 3

[MO14] MIRZA M., OSINDERO S.: Conditional generative adversarial nets. *Computer Science* (2014), 2672–2680. 2

[PKD*16] PATHAK D., KRAHENBUHL P., DONAHUE J., DARRELL T., EFROS A. A.: Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). 3

[RAY*16] REED S. E., AKATA Z., YAN X., LOGESWARAN L., SCHIELE B., LEE H.: Generative adversarial text to image synthesis. *International Conference on Machine Learning(ICML)* (2016), 1060–1069. 2

[RMC15] RADFORD A., METZ L., CHINTALA S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science* (2015). 5

[RMSC11] ROBSON C., MAHARIK R., SHEFFER A., CARR N.: Context-aware garment modeling from sketches. *Computers & Graphics(CG) 35*, 3 (2011), 604–613. 2

[SGZ*16] SALIMANS T., GOODFELLOW I., ZAREMBA W., CHEUNG V., RADFORD A., CHEN X.: Improved techniques for training gans. In *Advances in Neural Information Processing Systems* (2016), pp. 2234–2242. 9

[SLF*17] SANGKLOY P., LU J., FANG C., YU F., HAYS J.: Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 3

[SVI*16] SZEGEDY C., VANHOUCKE V., IOFFE S., SHLENS J., WOJNA Z.: Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2818–2826. 9

[TWB*07] TURQUIN E., WITHER J., BOISSIEUX L., CANI M.-P., HUGHES J. F.: A sketch-based interface for clothing virtual characters. *IEEE Computer graphics and applications(CGA) 27*, 1 (2007). 2

[VCMT05] VOLINO P., CORDIER F., MAGNENAT-THALMANN N.: From early virtual garment simulation to interactive fashion design. *Computer-aided design 37*, 6 (2005), 593–608. 2

[WLZ*18] WANG T.-C., LIU M.-Y., ZHU J.-Y., TAO A., KAUTZ J., CATANZARO B.: High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 2

[XSA*18] XIAN W., SANGKLOY P., AGRAWAL V., RAJ A., LU J., FANG C., YU F., HAYS J.: Texturegan: Controlling deep image synthesis with texture patches. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 2, 3, 4, 5, 9

[XT15] XIE S., TU Z.: Holistically-nested edge detection. *International Journal of Computer Vision 125*, 1-3 (2015), 1–16. 3

[YCYL*17] YEH R. A., CHEN C., YIAN LIM T., SCHWING A. G., HASEGAWA-JOHNSON M., DO M. N.: Semantic image inpainting with deep generative models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 3

[ZIE16] ZHANG R., ISOLA P., EFROS A. A.: Colorful image colorization. In *European Conference on Computer Vision* (2016), Springer, pp. 649–666. 2

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)* (Oct 2017). 2, 5, 9

[ZXL*17] ZHANG H., XU T., LI H., ZHANG S., HUANG X., WANG X., METAXAS D.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 5907–5915. 2

[ZZP*17] ZHU J.-Y., ZHANG R., PATHAK D., DARRELL T., EFROS A. A., WANG O., SHECHTMAN E.: Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems(NIPS)* (2017), Curran Associates, Inc., pp. 465–476. 2, 3, 5, 9, 10