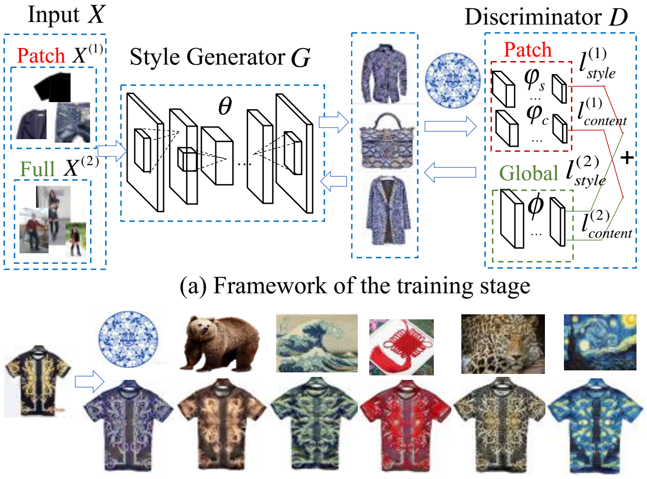
Fashion Style Generator

Shuhui Jiang[1] **and Yun Fu**[1,2]
[1]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA
[2]College of Computer and Information Science, Northeastern University, Boston, MA 02115, USA
{shjiang,yunfu}@ece.neu.edu

Abstract

In this paper, we focus on a new problem: applying artificial intelligence to automatically generate fashion style images. Given a basic clothing image and a fashion style image (e.g., leopard print), we generate a clothing image with the certain style in real time with a neural fashion style generator. Fashion style generation is related to recent artistic style transfer works, but has its own challenges. The synthetic image should preserve the similar design as the basic clothing, and meanwhile blend the new style pattern on the clothing. Neither existing global nor patch based neural style transfer methods could well solve these challenges. In this paper, we propose an end-to-end feed-forward neural network which consists of a fashion style generator and a discriminator. The global and patch based style and content losses calculated by the discriminator alternatively back-propagate the generator network and optimize it. The global optimization stage preserves the clothing form and design and the local optimization stage preserves the detailed style pattern. Extensive experiments show that our method outperforms the state-of-the-arts.



(a) Framework of the training stage

(b) Examples of fashion style generation

Figure 1: Fashion style generator framework overview. The input $\boldsymbol{X}$ consists of a set of clothing patches $\boldsymbol{X}^{(1)}$ and full clothing images $\boldsymbol{X}^{(2)}$. The system consists of two components: an image transformation network G served as fashion style generator, and a discriminator network D calculates both global and patch based content and style losses. G is a convolutional encoder decoder network parameterized by weights θ. Six generated shirts with different styles by our method are shown as examples. (We highly recommend to zoom in all the figures with color version for more details.)

1 Introduction

Applying artificial intelligence to solve problems in art and fashion fields attract a lot of attentions such as fashion style classification [FYihui Ma and Tong, 2017; Kiapour *et al.*, 2014; Jiang *et al.*, 2016a], clothing parsing [Yamaguchi *et al.*, 2013; Yamaguchi *et al.*, 2012], clothing retrieval [Jiang *et al.*, 2016b] and recommendation [Fu12 *et al.*, 2017]. In this paper, we focus on a novel problem: fashion style generation. It is different from existing online clothing design tools [1,2], which directly put a picked icon on the basic clothing. As shown in Figure 1 (b), with inputs of a basic clothing image and a style image, we automatically generate a clothing image blending with the new style while preserving the basic design. The definition of "style" in this paper is similar as the recent neural style transfer works [Gatys *et al.*, 2015]. Taking Van Gogh's "Starry Night" as the example style image, style is between the low-level color/texture (e.g., blue and yellow color, rough or smoother texture) and the high-level objects (e.g., house and mountain). "Style" is a relatively abstract concept. Fashion style generation has at least two practical usages. Designers could quickly see how the clothing looks like in a given style to facilitate the design processing. Shoppers could synthesize the clothing image with the ideal style and apply clothing retrieval tools [Jiang *et al.*, 2016b] to search the similar items.

Fashion style generation is related to existing neural style transfer works [Gatys *et al.*, 2015; Li and Wand, 2016a; Efros and Freeman, 2001], but has its own challenges. In fashion style generation, the synthetic clothing image should

[1] https://www.customink.com/lab?ref=nav_v2
[2] http://www.ooshirts.com

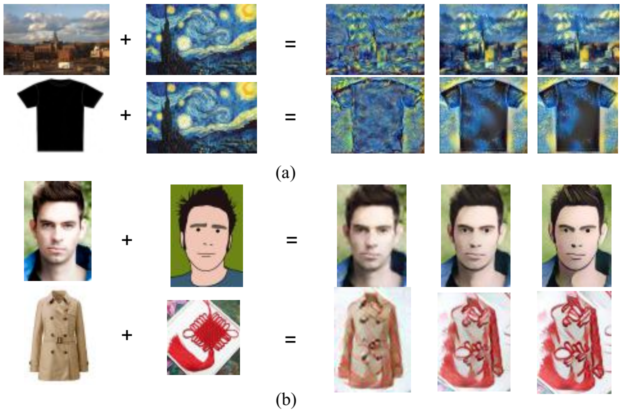


Figure 2: Limitations of applying the global [Gatys *et al.*, 2015] (a) and patch [Li and Wand, 2016a] based neural style transfer methods to fashion style generation. The left two columns are input content and style images. The right three columns are synthetic results in different iterations. In (a), we apply global method on artistic style transfer in the first row and on fashion style generation in the second row. In (b), we apply patch method on face-to-face transfer in the first row and on fashion style generation in the second row. This figure demonstrates that applying global or patch based methods may fail to synthesize high quality fashion style images.

blend the style of the style image while preserving the original form and shape of the clothing. Very few works have focused on fashion style generation. To our best knowledge, there is no publication so far and we only find an unpublished course project, which investigates Gatys's [Gatys *et al.*, 2016] neural style transfer work to fashion style transfer[3]. [Gatys *et al.*, 2016] performed artistic style transfer, combining the content of one image with the style of another by jointly minimizing the content reconstruction loss and the style reconstruction loss. Although [Gatys *et al.*, 2016] produces high quality results in painting style transfer, it is computationally expensive since each step of the optimization requires forward and backward passes through the pretrained network. Meanwhile, existing works are mainly focused on painting or other applications, which may not well capture the challenges of fashion style generation task.

Existing neural style transfer works mainly consist of two kinds of approaches: global and patch. Global (i.e., full image) based methods [Gatys *et al.*, 2015; Johnson *et al.*, 2016; Gatys *et al.*, 2016; Ulyanov *et al.*, 2016] achieve impressive results in artistic style transfer, but with limited fidelity in local detail, especially to high-resolution images. As shown in Figure 2 (a), the global structure of content images (i.e., buildings and T-shirt) is well preserved; however, the detailed structures of the style images are not well blended on the T-shirt. We could see that the yellow stars are transferred on the background instead of the T-shirt.

Patch based approaches, such as deep Markovian models [Li and Wand, 2016a; Li and Wand, 2016b; Ding *et al.*, 2016], capture the statistics of local patches and assemble them to high-resolution images. While they achieve high fidelity of details, the additional guidance is required if the global structure should be reproduced [Efros and Freeman, 2001; Li and Wand, 2016a; Li and Wand, 2016b]. As shown in Figure 2 (b), patch based approaches well preserve both global and local structure only when the style and content images are with the similar structure such as face-to-face. However, in fashion style generation, the style image is not necessarily to be the clothing image or with the similar structure as the content image. Lack of additional global guidance would destroy the global structure of the synthetic image. For example, in the second row of Figure 2 (b), the global structure of the left part of the synthetic clothing is destroyed during the synthesis processing.

To address the above challenges, we propose an end-to-end feed-forward neural network of fashion style generation. We combine the benefits of both global and patch based methods, and meanwhile avoid the disadvantages. As shown in Figure 1, the inputs consist of a set of clothing patches and full images. There are two components: an image transformation network G served as the fashion style generator, and a discriminator network D calculates both global and patch based content and style reconstruction losses. Furthermore, an alternating global-patch back-propagation strategy is proposed to optimize the generator to preserve both global and local structures. In online generation stage, we only need to do the forward propagation, which makes it is hundreds faster than the existing methods with both forward and backward passes [Li and Wand, 2016a; Gatys *et al.*, 2016]. Experimental results demonstrate that for both speed and quality, the proposed method outperforms the state-of-the-arts in fashion style generation task.

2 Method

2.1 Problem Formulation

For an input clothing image q and a style image y_s, we want to synthesize a clothing image $\hat{y}$ through a style generator G. $\hat{y}$ blends the style of y_s on q and meanwhile preserves the form and design of q. We achieve it through off-line training the parameters θ of G with a set of clothing images $\boldsymbol{X}$ and the style image y_s.

Recently, a wide variety of feed-forward image transformation tasks have been solved by training deep convolutional neural networks [Johnson *et al.*, 2016; Li and Wand, 2016b]. A general feed-forward network consists of an image transformation network G and a discriminator network D. For style transfer/generation, G is served as the a style generator. The reconstruction content and style loss of D iteratively back-propagates and optimizes θ. In online generation, G transforms the input clothing image q into output clothing image $\hat{y}$ via the mapping $\hat{y} = f_\theta(q)$. Thus, we do not need to do back-propagation, which facilitates the real time generation.

However, as discussed above, neither the existing global [Johnson *et al.*, 2016] nor patch [Li and Wand, 2016b] based methods could well solve the challenges in fashion style generation. Therefore, we propose to jointly consider the global and patch reconstruction losses when optimizing G to overcome the shortcomings of global or patch based methods. The

[3] http://personal.ie.cuhk.edu.hk/~lz013/papers/fashionstyle_poster.pdf

main purpose of global based optimization is to preserve the global form and design of the basic clothing, while the main purpose of patch based optimization is to preserve the local details of the style pattern.

2.2 Architecture

The flowchart of Figure 1 shows the training stage of our system. Different from existing works either only use full images or patches, the input $\boldsymbol{X}$ of our training stage consists of a set of clothing patches $\boldsymbol{X}^{(1)}$ and full clothing images $\boldsymbol{X}^{(2)}$. $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ are applied in patch and global based optimization stage respectively. The patch images are cropped from the online shopping clothing dataset [Hadi Kiapour *et al.*, 2015; Jiang *et al.*, 2016b]. They are usually with clean backgrounds and front poses, which makes it much easier to focus on the details of the local clothing structure. The whole clothing images are from the Fashion 144k dataset [Simo-Serra and Ishikawa, 2016]. They are usually with complex backgrounds and different poses, which makes the model more robust to noise and could well preserve the global clothing structure.

Our system is an end-to-end feed-forward neural network consists of an image transformation network G with parameter θ served as the fashion style generator and a discriminator network D. G consists of encoder and decoder parts. The encoder E_n encodes the input image as a vector and decoder D_e decodes the vector again as an image. D consists of the global loss network ϕ and the patch loss network φ_s and φ_c for style and content respectively. The reconstruction loss back-propagates and optimizes θ to make the synthesis image preserves both global structure and local details.

As mentioned in [Johnson *et al.*, 2016], the pretrained convolutional neural networks are able to extract perceptual information and encode semantics. Therefore, we utilize a pretrained image classification network (i.e., VGG-19) [Simonyan and Zisserman, 2014; Li *et al.*, 2016] as the initialization of E_n. Also, the VGG network is utilized as the global loss network ϕ and the patch content loss network φ_c.

For the patch style loss network φ_s, since existing network are mainly trained for whole images, instead of directly applying an existing pretrained discriminator network, we apply the generative adversarial training [Goodfellow *et al.*, 2014] for learning the parameters of φ_s and initializing D_e simultaneously. After the initialization, an alternating patch-global training strategy is applied for optimizing the generator parameter θ.

2.3 Objective Function of Discriminator

As discussed above, the loss function L of the discriminator D is defined as a weighted combination of the patch based loss $L^{(1)}$ and the global based loss $L^{(2)}$:

$$\begin{aligned} L(\hat{y}, y_c, y_s) &= L^{(1)}(\hat{y}, y_c, y_s) + \lambda L^{(2)}(\hat{y}, y_c, y_s) \\ &= \underbrace{l^{(1)}_{\text{style}} + \lambda_1 l^{(1)}_{\text{content}}}_{\text{patch}} + \underbrace{\lambda_2 l^{(2)}_{\text{style}} + \lambda_3 l^{(2)}_{\text{content}}}_{\text{local}}, \end{aligned} \quad (1)$$

where λ, λ_1, λ_2 and λ_3 are tuning parameters to adjust the weights. Given an input training clothing image $x \in \boldsymbol{X}$, $\hat{y}$ is the output synthetic image of the generator through mapping $\hat{y} = f_\theta(x)$. y_s is the input style image. y_c is the clothing content image. In the patch optimization stage, $y_c = x \in \boldsymbol{X}^{(1)}$, while in global optimization stage, y_c is a higher resolution version of the image $x \in \boldsymbol{X}^{(2)}$.

Both $L^{(1)}$ and $L^{(2)}$ consist of two parts of losses: the content and the style reconstruction loss. The content losses $l^{(1)}_{\text{content}}(\hat{y}, y_c)$ and $l^{(2)}_{\text{content}}(\hat{y}, y_c)$ capture the distances in respect of perceptual features between y_c and $\hat{y}$, for patch and global respectively. The style losses $l^{(1)}_{\text{style}}(\hat{y}, y_s)$ and $l^{(2)}_{\text{style}}(\hat{y}, y_s)$ capture the distances between mid-level features of y_s and $\hat{y}$ for patch and global respectively. In the following, we introduce $l^{(2)}_{\text{content}}$, $l^{(2)}_{\text{style}}$, $l^{(1)}_{\text{content}}$, and $l^{(1)}_{\text{style}}$ one by one.

As discussed above, we apply a pretrained convolutional neural networks (i.e., VGG-19) as the global loss network ϕ. The deeper layers of ϕ extract perceptual information and encode semantics of the content. Thus, measuring the perceptual similarity of y_c and $\hat{y}$ as the content loss is more informative than encouraging the pixel-based match. The middle layers of ϕ, instead, extract mid-level feature representation as the image style. Thus we measure the middle layer similarity of y_s and $\hat{y}$ as the style loss. Let ϕ_j and ϕ_k be the activations of the j-th (deeper) and k-th (middle) layer of the network ϕ. $C_j \times H_j \times W_j$ is the shape of feature map of the j-th layer. In order to make the output image in the high resolution, we assign y_c as the higher resolution version of the input image $x \in \boldsymbol{X}^{(2)}$. $l_{\text{content}}(\hat{y}, y_c)$ is the Euclidean distance between feature representation as:

$$l^{(2)}_{\text{content}}(\hat{y}, y_c) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y_c)\|^2_2, \quad (2)$$

and for global style loss, we use the Frobenius norm of differences of the Gram matrices [Gatys *et al.*, 2015]:

$$l^{(2)}_{\text{style}}(\hat{y}, y_s) = \frac{1}{C_k H_k W_k} \|Gram^{\phi}_k(\hat{y}) - Gram^{\phi}_k(y_s)\|^2_F. \quad (3)$$

Different from $l^{(2)}_{\text{content}}$ and $l^{(2)}_{\text{style}}$ computed on the same loss network ϕ, patch losses $l^{(1)}_{\text{content}}$ and $l^{(1)}_{\text{style}}$ are computed on patch content loss network φ_c and patch style loss network φ_s respectively. Assume we extract N patches from a full image and denote $\Psi(\cdot)$ as the patches extracted from the image. For content loss, we calculate the Euclidean distance between feature representation in the similar way as Eq. (2):

$$l^{(1)}_{\text{content}}(\hat{y}, y_c) = \frac{1}{N} \|\varphi_c(\Psi(\hat{y}) - \varphi_c(\Psi(y_c)\|^2_2, \quad (4)$$

where $\Psi(\hat{y})$ and $\Psi(y_c)$ are patches extracted from $\hat{y}$ and y_c.

For patch style loss network φ_s, since existing networks are mainly trained for full images, instead of directly applying the existing pretrained discriminator network, we apply Generative Adversarial Network (GAN) [Goodfellow *et al.*, 2014; Radford *et al.*, 2015] for learning φ_s and meanwhile initializing the parameters of decoder D_e of the generator. We will describe it in the next subsection. After obtaining the φ_s, we apply Hinge loss to measure the style loss as [Li and Wand, 2016b]:

$$l_{\text{style}}(\hat{y}, y_s) = \frac{1}{N}\sum_{i=1}^{N}\max(0, 1 - 1 \times s_i), \tag{5}$$

where s_i denotes the classification score of i-th neural patch. More details could be referred in [Li and Wand, 2016b].

2.4 Optimization of Generator

In this section, we describe the strategy to optimize the parameter θ of the style generator G using the loss L calculated by the discriminator:

$$\theta^* \leftarrow \arg\min_{\theta} \mathbb{E}_{x,y_s,y_c}[L\big(f_\theta(x), y_s, y_c\big)], \tag{6}$$

where $\mathbb{E}_{x,y_s,y_c}$ is the estimation of the expectation via the training set $\{x, y_s, y_c\}$, $x \in \boldsymbol{X}$.

We firstly describe utilizing GAN [Goodfellow *et al.*, 2014; Radford *et al.*, 2015] for learning patch style network φ_s and meanwhile initializing the parameters of decoder D_e. The inputs of this stage are image patches $\boldsymbol{X}^{(2)}$ and the style image y_s. As described, the parameters of the E_n, the global loss net ϕ and the local content loss net φ_c are initialized by VGG. We keep E_n unchanged in this step.

GAN estimates generative models via an adversarial process. The training procedure for G is to maximize the probability of D making a mistake. The objective function is as:

$$\begin{aligned}\min_G \max_D V(D, G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)]\\ +\mathbb{E}_{z\sim p_z(z)}[log(1 - D(G(z)))].\end{aligned} \tag{7}$$

In traditional GAN, z is the random noise. In our work, we replace z using the encoded feature of the input image by E_n of VAE [Kingma and Welling, 2013]. The detailed theory proof could be referred in [Goodfellow *et al.*, 2014]. Figure 3 shows three examples of the generated patches with the style "Chinese knot" after the initialization of φ_s and D_e. To this end, all the parts of networks are initialized.

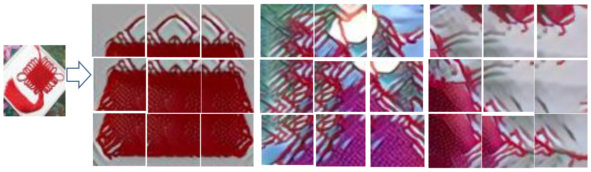


Figure 3: Example of generated style patches. The inputs are image patches and a style image "Chinese knot". We could see that the generator blends the style of "Chinese knot" on the clothing patches detailedly.

Next, we describe the alternating global-patch back-propagation algorithm for optimizing θ. The discriminator networks are unchanged during the optimization. The alternating global-patch back-propagation iterates the following two-steps for T iterations.

(1)*Global back-propagation*:

In the global back-propagation step, θ_{t+1} can be obtained by using the least squares error of the global loss in iteration $t + 1$ and t as $\|L^{(2)}_{t+1} - L^{(2)}_t\| = \|e^{(2)}_{m+1}\|$ to train the generator $f_\theta(x)$. We employ a gradient descent (GD) algorithm to minimize $\|e_{m+1}\|$. θ_{t+1} is updated by repeating $\tau^{(2)}$ times as:

Algorithm 1 Alternating Patch-Global Back-propagation

INPUT: $\boldsymbol{X}^{(1)}$, $\boldsymbol{X}^{(2)}$, y_s, T, $\tau^{(1)}$, $\tau^{(2)}$. VGG network parameter.

```
Initialize weights of $E_n$, $\phi$, $\varphi_c$ by VGG.
Apply GAN to initialize $D_e$ and $\varphi_s$.
for $t$=1,2,...,$T$ do
    %update $\theta$ by global loss back-propagation.
    for $m$=1,2,...,$\tau^{(2)}$ do
        Calculate the global loss by Eq. (1),(2),(3).
        Update $\theta_t$ by Eq. (8).
    end for
    %update $\theta$ by patch loss back-propagation.
    for $m$=1,2,...,$\tau^{(1)}$ do
        Calculate the patch loss by Eq. (1),(4),(5).
        Update $\theta_t$ by Eq. (9).
    end for
    Update $\theta_{t+1} = \theta_t$.
end for
ONPUT: Style generator parameter $\hat{\theta} = \theta_t$.
```

$$\theta_t = \theta_t - \eta^{(2)}\frac{\partial\|e^{(2)}_{t+1}\|^2_2}{\partial\theta_t}, \tag{8}$$

where $\eta^{(2)}$ is the learning rate.

(2)*Patch back-propagation*:

In local back-propagation step, θ_{t+1} can be obtained by using the least squares error of the patch loss in iteration $t+1$ and t as $\|L^{(1)}_{t+1} - L^{(1)}_t\| = \|e^{(1)}_{m+1}\|$ to train the generator $f_{\theta(x)}$. θ_{t+1} is updated by repeating $\tau^{(1)}$ times as:

$$\theta_t = \theta_t - \eta^{(1)}\frac{\partial\|e^{(1)}_{t+1}\|^2_2}{\partial\theta_t} \tag{9}$$

where $\eta^{(1)}$ is the learning rate.

The algorithm of optimization is described in Algorithm 1.

3 Experiments

3.1 Experimental Details

Dataset and Data Processing: Our training dataset contains two parts: A Fashion 144k dataset as full image inputs [Simo-Serra and Ishikawa, 2016] and 300 online shopping images as patch inputs, which are randomly selected from the Online Shopping dataset [Hadi Kiapour *et al.*, 2015]. Existing patch based works point out that only a small number of training images (i.e., 100 images) could still produce good results [Li and Wand, 2016b]. The Fashion 144k dataset consists of 144,169 user posts with images, collected from the largest fashion website chictopia.com. The Online Shopping dataset consists of 404,683 shop photos from 25 different online clothing retailers. Our testing data are 100 images randomly collected from online shopping websites. In the exper-

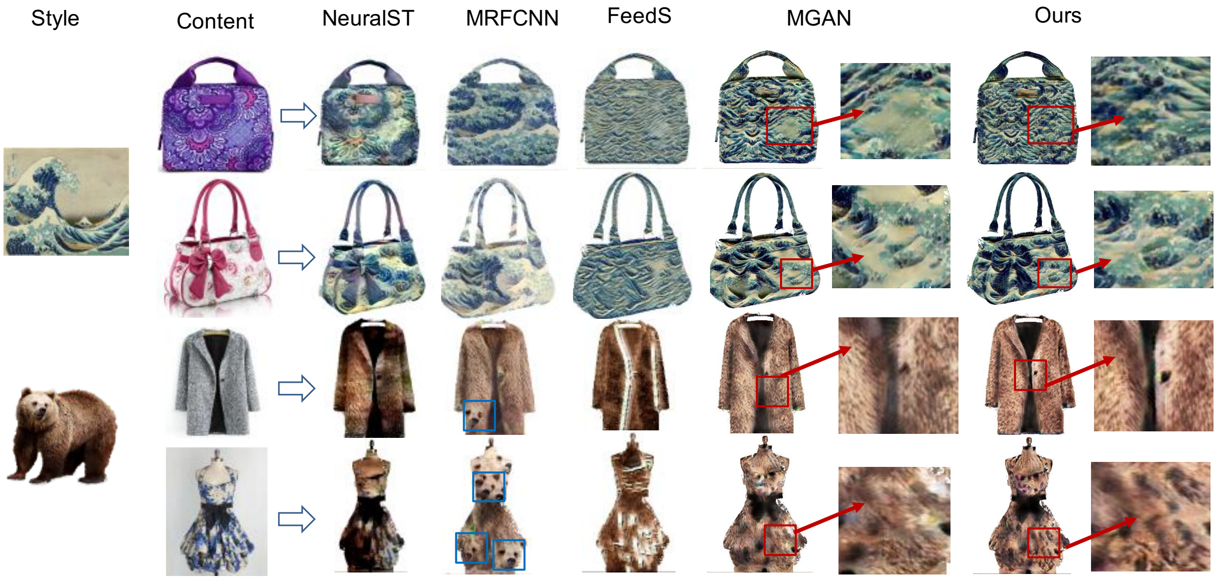


Figure 4: Synthetic fashion style images by 5 compared methods NeuralST, MRFCNN, FeedS, MGAN and Ours. The fist left column shows the input style images "wave" and "bear". The second left column shows four input content images. For MGAN and Ours, we enlarge the regions in red frames to show more details.

iments, we apply 6 style images as shown in the last second row in Figure 1. They are "blue and white porcelain", "bear", "wave", "Chinese knot", "leopard print" and "starry night".

The settings of the sizes of inputs and outputs images in training are following existing global and patch based works [Johnson *et al.*, 2016; Li and Wand, 2016b]. The style images are color images of shape $3 \times 256 \times 256$. For full images, the low-resolution inputs are of shape $3 \times 72 \times 72$. The high-resolution inputs are of shape $3 \times 288 \times 288$. For patch images, the patches are of shape $3 \times 128 \times 128$. They are cropped from full online shopping images with a fixed stride, which is 16 in our work. Since the image transformation networks are fully-convolutional, at test stage they can be applied to images of any resolution.

Network details: For the generator network G, it takes a VGG_19 layer $relu4\_1$ encoding of an image and directly decodes it to pixels of the synthesis image. For the decoder D_e and the patch style loss network φ_s , like [Radford *et al.*, 2015; Wu *et al.*, 2016], we use batch normalization (BN) and LReLU to improve the training. The style loss is computed at the VGG_19 network layer $relu2\_2$, and the content loss is computed in VGG_19 layer $relu5\_1$.

Training details: For global stage back-propagation, maximum iteration is set to be 40000, and a batch size of 4 is applied. These settings give roughly 1.5 epochs over all the training data. For patch stage back-propagations, we test 1 to 10 epochs over all the patches. The optimization is based on Adam [Kingma and Ba, 2014] with a learning rate of 1×10^{-3}. No weight decay or dropout is used. The training is implemented using Torch [Collobert *et al.*, 2011] and cuDNN [Chetlur *et al.*, 2014]. Each style training takes around 7 hours on a single GTX Titan X GPU.

3.2 Compared Methods

Although there are very few publications fully focused on fashion style generation task, to evaluate the effectiveness of our proposed method, we take four most related global or patch based neural style transfer works as our baseline methods as following:

NeuralST [Gatys *et al.*, 2015]: Gatys et al. performed artistic neural style transfer by synthesizing a new image that matches both the content of the content image and the style of the style image.

MRFCNN [Li and Wand, 2016a]: Li et al. combined generative Markov random field (MRF) patch based models and discriminatingly trained deep convolutional neural networks (dCNNs) for synthesizing 2D images.

FeedS [Johnson *et al.*, 2016]: Johnson et al. proposed feed-forward network to solve the optimization problem in [Gatys *et al.*, 2015] in real time in test stage.

MGAN [Li and Wand, 2016b]: Li et al. proposed a Markovian patch-based feed-forward network for artistic style transfer. This work is similar as the initialization of the patch loss network in our work.

Ours: It includes the whole pipeline of our framework.

In NeuralST and MRFCNN, both forward and backward propagations are applied when generating testing results. For FeedS and MGAN, we train the feed-forward networks with the same clothing datasets as our work. We have conducted different settings of parameters and post the best results we obtained of each method. For the comparison methods, we run the code released by the authors.

3.3 Experimental Results

Figure 4 compares our results with compared methods NeuralST, MRFCNN, FeedS and MGAN. In NeuralST and MRFCNN, we set the iteration number as 200. In FeedS, we set

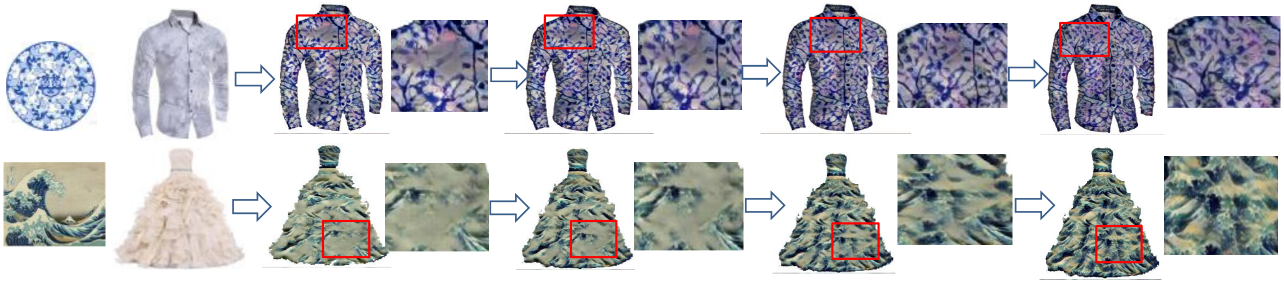


Figure 5: Illustration of synthetic clothings at different iterations (0, 1000, 2000, 3000 from left to right) of the global back-propagation after the patch based initialization. The global iterations gradually add the style pattern on the destroyed parts of the images caused by the patch initialization. We enlarge the parts in the red frames to show more details.

the iteration number as 40,000, which is almost 2.5 epochs. In MGAN, we set the iteration number as 3000, which is almost 10 epochs. In Ours, we set $T = 1$ and $\tau^{(1)} = \tau^{(2)} = 3000$. We remove the backgrounds of clothing images through image matting algorithms for better visualization.

When comparing feed-forward based methods (FeedS, MGAN and Ours), we found that MGAN and Ours better preserve the detailed textures in the style images, compared with global based FeedS. For example, the claws of the waves and bear hair are very clear. Since our network is initialized by patch based network, the difference of the texture between MGAN and Ours are not large. However, as discussed above, patch based methods may not well preserve the global structure of the full image. For example, in the first row of MGAN, the areas in the red frames are not well synthesized. In our method, these areas are better blended with style patterns. It shows the effectiveness of considering both global and local characteristics in our method.

NeuralST and MRFCNN are not feed-forward based networks. Generally, besides the speed, we have the similar observations. In MRFCNN, although the generated images preserve the textures, they may loss the original global structures. For example, on the two generated images with bear style in MRFCNN, even the head of bears are transferred.

3.4 Discussion of Speed and Complexity

NeuralST and RMFCNN are computationally expensive since each step of the optimization requires forward and backward passes through the pretrained network. With the feed-forward network, since we do need to do the back-propagation in the test stage, the test speed is hundreds faster.

For the training stage, the most time-consuming part is the patch discriminator network initialized by GAN. The time complexity of this step is the same as [Li and Wand, 2016b]. It is mainly effected by the training iterations and the batch size. In our work, it take about 5 hours for the initialization. After initialization, the speed is effected by the alternating iteration number T, and the iteration numbers $\tau^{(1)}$ and $\tau^{(2)}$ in the patch and global back-propagation. Since the generator is already initialized, we set T, $\tau^{(1)}$ and $\tau^{(2)}$ at small numbers. It takes about 2 hours for the following optimization.

3.5 Discussion of Our Method

To evaluate the effectiveness of the alternating patch-global back-propagation, in Figure 5, we show the generated images of only utilizing the patch back-propagation (iteration 0) and after global back-propagation iterations at 1000, 2000 and 3000. The global back-propagation gradually blends the style on the destroyed parts caused by the patch initialization, which shows the effectiveness of the patch-global optimization strategy.

We also discuss the weight λ in our objective function Eq. (1). We tune λ through different settings of learning rate $\eta^{(1)}$ and $\eta^{(2)}$ in Eq. (8) and (9). The initial learning rate $\eta^{(1)}$ in patch optimization is 0.02. We fix $\eta^{(1)}$ and tune $\eta^{(2)}$ of global optimization as e^{-5} to e^{-9}. If we set the learning rate too large, the network could not be converged and the output image would be blur and without style patterns blended. We achieve good results at $\eta^{(2)}$ around e^{-7}. Comparing $\eta^{(1)}$ and $\eta^{(2)}$, we observed that the patch loss plays an more important role than global loss.

3.6 Limitation

Our work still has some limitations. First, similar as the patch based method MGAN [Li and Wand, 2016b], we may also fail to generate style texture on the clothing if a very large area of image is non-texture and pain. Second, sometimes the color would be less accurate, due to the network may preserve some original color of the content image. Third, the resolution of the generated clothings are still lower than the real clothing.

4 Conclusion

In this paper, we focused on fashion style generation, which is a relatively new topic in artificial intelligence field. We pointed out the challenges in fashion style generation compared with existing artistic neural style transfer. The synthetic image should preserve the similar design as the basic clothing and meanwhile blend the detailed style. We analyzed the shortcomings of existing global and local methods in neural style transfer if directly applied in our task. To address the challenges, we proposed an end-to-end neural fashion style generator, together with an alternating patch-global back-propagation strategy. Experiments and analysis show that our model outperforms the state-of-the-arts.

References

[Chetlur *et al.*, 2014] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

[Collobert *et al.*, 2011] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[Ding *et al.*, 2016] Zhengming Ding, Ming Shao, and Yun Fu. Deep robust encoder through locality preserving low-rank dictionary. In *Proceedings of ECCV*, pages 567–582. Springer, 2016.

[Efros and Freeman, 2001] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[Fu12 *et al.*, 2017] Jingtian Fu12, Jia Jia, Yihui Ma, Fanhang Meng, and Huan Huang. A virtual personal fashion consultant: Learning from the personal preference of fashion. In *Proceedings of AAAI*. AAAI Press, 2017.

[FYihui Ma and Tong, 2017] Suping Zhou Jingtian Fu Yejun Liu FYihui Ma, Jia Jia and Zijian Tong. Towards better understanding the clothing fashion styles: A multimodal deep learning approach. In *Proceedings of AAAI*. AAAI Press, 2017.

[Gatys *et al.*, 2015] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of CVPR*, pages 2414–2423, 2016.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of NIPS*, pages 2672–2680, 2014.

[Hadi Kiapour *et al.*, 2015] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. In *Proceedings of ICCV*, pages 3343–3351, 2015.

[Jiang *et al.*, 2016a] Shuhui Jiang, Ming Shao, Chengcheng Jia, and Yun Fu. Consensus style centralizing auto-encoder for weak style classification. In *Proceedings of AAAI*, pages 1223–1229. AAAI Press, 2016.

[Jiang *et al.*, 2016b] Shuhui Jiang, Yue Wu, and Yun Fu. Deep bi-directional cross-triplet embedding for cross-domain clothing retrieval. In *Proceedings of MM*, pages 52–56. ACM, 2016.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155*, 2016.

[Kiapour *et al.*, 2014] M Hadi Kiapour, Kota Yamaguchi, Alexander C Berg, and Tamara L Berg. Hipster wars: Discovering elements of fashion styles. In *Proceedings of ECCV*, pages 472–488. Springer, 2014.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Li and Wand, 2016a] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. *arXiv preprint arXiv:1601.04589*, 2016.

[Li and Wand, 2016b] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *arXiv preprint arXiv:1604.04382*, 2016.

[Li *et al.*, 2016] J. Li, , T. Zhang, W. Luo, J. Yang, X.T. Yuan, and J. Zhang. Sparseness analysis in the pertraining of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2016.2541681, 2016.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Simo-Serra and Ishikawa, 2016] Edgar Simo-Serra and Hiroshi Ishikawa. Fashion style in 128 floats: joint ranking and classification using weak data for feature extraction. In *Proceedings of CVPR*, pages 298–307, 2016.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Ulyanov *et al.*, 2016] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of ICML*, 2016.

[Wu *et al.*, 2016] Yue Wu, Jun Li, Yu Kong, and Yun Fu. Deep convolutional neural network with independent softmax for large scale face recognition. In *Proceedings of MM*, pages 1063–1067. ACM, 2016.

[Yamaguchi *et al.*, 2012] Kota Yamaguchi, M Hadi Kiapour, Luis E Ortiz, and Tamara L Berg. Parsing clothing in fashion photographs. In *Proceedings of CVPR*, pages 3570–3577. IEEE, 2012.

[Yamaguchi *et al.*, 2013] Kota Yamaguchi, M Hadi Kiapour, and Tamara L Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *Proceedings of ICCV*, pages 3519–3526, 2013.