Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8894309, 12 pages
https://doi.org/10.1155/2020/8894309

Research Article

Image Localized Style Transfer to Design Clothes Based on CNN and Interactive Segmentation

Hanying Wang,[1] Haitao Xiong ,[2] and Yuanyuan Cai[1]

[1]*School of E-Business and Logistics, Beijing Technology and Business University, Beijing 100048, China*
[2]*School of International Economics and Management, Beijing Technology and Business University, Beijing 100048, China*

Correspondence should be addressed to Haitao Xiong; xionghaitao@btbu.edu.cn

Received 17 August 2020; Revised 6 December 2020; Accepted 14 December 2020; Published 29 December 2020

Academic Editor: Silvia Conforto

In recent years, image style transfer has been greatly improved by using deep learning technology. However, when directly applied to clothing style transfer, the current methods cannot allow the users to self-control the local transfer position of an image, such as separating specific T-shirt or trousers from a figure, and cannot achieve the perfect preservation of clothing shape. Therefore, this paper proposes an interactive image localized style transfer method especially for clothes. We introduce additional image called outline image, which is extracted from content image by interactive algorithm. The interaction consists simply of dragging a rectangle around the desired clothing. Then, we introduce an outline loss function based on distance transform of the outline image, which can achieve the perfect preservation of clothing shape. In order to smooth and denoise the boundary region, total variation regularization is employed. The proposed method constrains that the new style is generated only in the desired clothing part rather than the whole image including background. Therefore, in our new generated images, the original clothing shape can be reserved perfectly. Experiment results show impressive generated clothing images and demonstrate that this is a good approach to design clothes.

1. Introduction

In recent years, convolutional neural network has successfully completed a series of computer vision tasks, such as object detection, object recognition, image segmentation, and texture synthesis. In 2012, Krizhevsky et al. trained a large deep convolutional neural network and significantly improved the object recognition capability in the ImageNet challenge [1]. This has promoted many research studies and developments in the field of fashion, such as clothing classification and retrieval, clothing parsing, and recommendation. Yamaguchi et al. demonstrated an effective method for parsing clothing in fashion photographs and provided a large novel dataset and tools for labelling garment items to enable future research on clothing estimation [2]. Kalantidis et al. presented a scalable approach to automatically suggest relevant clothing products [3]. Liu et al. proposed a new deep model, namely, FashionNet, which learns clothing features by jointly predicting clothing attributes and landmarks [4]. The FashionNet develops powerful algorithms in clothes recognition and facilitates future research. Ma et al. proposed a novel fashion-oriented multimodal deep learning based model, bimodal correlative deep autoencoder (BCDA), to capture the internal correlation in clothing collocations [5].

In this paper, we focus on another purpose: the self-control of clothing style in local position. According to the trend of fashion, users select a clothing image as content image from Internet or shopping mall, and then find an art or a picture that they appreciate as a style image. Then, this algorithm is adopted to generate a unique clothing design, which combines the style of the style image and the clothing shape of content image. The term content refers to the shape of clothes, and the term style refers to patterns or colour in an image. In order to achieve the perfect preservation of clothing shape, we introduce a third image called outline image, which is extracted from content image by interactive GrabCut algorithm. The interaction consists simply of dragging a rectangle around the desired clothing. Then, we introduce an outline loss function based on distance

transform of the outline image. In order to smooth and denoise the boundary region, total variation is employed. The proposed method allows users to interactively control the location of image style transfer and constrains that the new style is generated only in the desired clothing part rather than the whole image including background, which not only retains the basic shape of the original clothing but also designs a new style for it.

The main contributions are as follows. (1) People often encounter the situation that they like the shape of clothes but do not like the pattern or colour on the clothes. This research can customize styles for the same clothes to meet the fashion needs of customers. (2) The embarrassing scene of wearing the same dress is always annoying, but the expensive high-level customization is not available to every ordinary person. This research allows ordinary users who are not professional designers to easily design their own clothes, with low cost and high satisfaction of pursuing uniqueness. (3) This research can provide a lot of inspiration for professional designers. With the style given by style image, the design can be completed quickly, which improves efficiency and customer satisfaction.

2. Related Work

In the field of computer vision based on statistics, many scholars have begun to study image style transfer. Image style transfer can be considered as a process of image rendering, which is generally regarded as an extension of texture synthesis. Previous texture modelling methods mainly focus on parametric texture modelling with summary statistics [6–8] and nonparametric texture modelling with MRFs [9, 10]. The former one is to capture image statistics from a sample texture and exploit summary statistical property to model the texture [11]. The latter one is to use nonparametric resampling. A variety of nonparametric methods are based on the MRF model, which assumes that in a texture image, each pixel is entirely characterized by its spatial neighbourhood [11].

In 2015, Gatys et al. [12] proposed the first neural algorithm of artistic style, which can separate and recombine the content and style of natural images. They designed a model based on convolutional neural network, using pretrained VGG network to extract and store the features of images. Then, the total loss, which is a linear combination between the content and the style loss, is updated by backpropagation until the new image simultaneously matches the style features of the style image and the content features of the content image. Johnson et al. introduced perceptual loss functions for training feed-forward networks for image transfer tasks and greatly improved transfer speed [13]. Li et al. proposed a novel interpretation of neural style transfer by treating it as a domain adaptation problem and argued that the essence of neural style transfer is to match the feature distributions between the style images and the generated images [14]. However, these algorithms have some problems, such as the complexity of manual modulation and the instability of Gram matrix during optimization. Therefore, Chen et al. proposed StyleBank, which is composed of multiple convolution filter banks and each filter bank explicitly represents one style, for neural image style transfer [15]. This method, which runs in real time, is easy to train and produces results that are qualitatively better. Zhang and Dana found it fundamentally difficult to finish comprehensive style modelling using 1-dimensional style embedding. Therefore, they introduced CoMatch Layer that learns to match the second-order feature statistics with the target styles and built a multistyle generative network (MSG-Net), which achieves real-time performance [16].

The above algorithms achieve impressive results in style transfer, but with limited fidelity in local details, lacking good retention of content details in the process of style transfer and lacking semantic and depth information contained in content images. If applied to the fashion style transfer for clothing, the resolution of the generated clothing image will be very low, the shape of the clothing will be deformed, and the colour of the original clothing will be retained, which is difficult to combine with the new style. In order to achieve high fidelity of details, some scholars adopted patch-based approaches to image style transfer. Chen and Schmidt proposed a simpler optimization objective based on local matching that combines the content structure and style textures in a single layer of the pretrained network and then conducted the style swap between the content and the style [17]. Li and Wand introduced Markovian generative adversarial networks (MGANs) to capture the statistics of local patches and assemble them to high-resolution images [18]. Then, they combined generative Markov random field (MRF) models with discriminatively trained deep convolutional neural networks (dCNNs), which can both match and adapt local features with considerable variability [19].

Recently, deep learning methods have shown superior performance in analysis images [20–22]. Specially, Jiang and Fu introduced GAN to style transfer to address the challenge [23]. However, 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. It is difficult for style image to have the similar structure as the clothing image. They proposed an end-to-end feed-forward neural network which includes a fashion style generator and a discriminator. They combined global and patch-based methods, and the inputs included a set of clothing patches and full images. The clothing shape and design are preserved by the global optimization stage, and the detailed style pattern is preserved by the local optimization stage. Experiments and analysis show that their generated images are impressive and outperform other algorithms. However, their work still has some limitations. If a large area of clothing is plain and nontexture or does not have any patterns, the style texture on the clothing may fail to be generated. Furthermore, the generation of method based on GAN is not stable. The new image has high resolution only when spanning space is well constrained. In additional, generative adversarial network, as a data-driven method, has to obtain plenty of data. When applied to fashion style transfer, it is very inconvenient for ordinary users to design their own new clothes.

Different from other research studies, this paper applies the neural style transfer to fashion field, enabling ordinary

users to design their own unique clothing according to their preferences. The input of this approach can be a clothing image with complex background rather than a simple clothing image without person and background. The paper focuses on users controlling the local transfer position of an image, facilitating the operation and greatly making it convenient for general public to design a clothing style. The research aims to achieve the perfect preservation of clothing shape, obtaining a new clothing design that simultaneously matches the style of the style image and the clothing shape of the content image.

3. Image Style Transfer

Figure 1 shows the process of image style transfer in this paper. (1) Using pretrained VGG network to extract and store the features of content and style images. (2) Obtaining an outline image by using GrabCut algorithm to segment content image, by simply dragging a rectangle around the desired clothing. (3) Obtaining the distance transform matrix by the distance transform of the outline image. (4) Obtaining outline feature by taking pixelwise power of distance transform matrix to emphasize the specific group of pixels. (5) Calculating content, style, and outline loss based on their features. (6) Adding total variation to total loss, which is a linear combination between the content, style, and outline loss.

3.1. Image Segmentation. An image is divided or partitioned into various parts called segments based on its colour, grey, texture, and other features so that these features show differences between different areas but show similarities in the same areas. At present, the main image segmentation algorithms include segmentation based on region, which divides the objects into different parts based on similarity of grey distribution, segmentation based on threshold, which divides the objects into different parts based on grayscale of pixels, segmentation based on edge which uses an image's discontinuous local features to detect edges and thus define a boundary of the object, and segmentation based on clustering which separates the pixels of the image into homogeneous clusters.

The above algorithms all have been widely used for image segmentation in different fields. They are simple but powerful methods for separate objects from background. However, when there are several similar targets in the same background, such as one person with T-shirt and trousers, they will separate the whole figure from background rather than only T-shirt or only trousers. Therefore, in order to make users specifically control the local transfer position of an image and to make the operation as simple as possible, this paper adopts interactive GrabCut algorithm [24] in the handling of segmentation. GrabCut algorithm is an improvement on GraphCut algorithm [25]. GraphCut algorithm needs to mark both "object" and "background" segments. Hard constraints for segmentation are imposed by indicating certain pixels that absolutely have to be part of the object and certain pixels that have to be part of the background [25]. But GrabCut algorithm needs significantly fewer interactions. The interaction consists simply of dragging a rectangle around the specific clothing that requires style transfer. Pixels outside the rectangle are marked as known background, and pixels inside are marked as unknown, which imposes certain hard constraints on segmentation. Then, a model is created to determine whether the unknown pixels are foreground or background.

3.1.1. The GrabCut Segmentation Algorithm. The image is regarded as an array $Z = (z_1, \ldots, z_n, \ldots, z_N)$ of grey values in graph cut algorithm, and this array is used to describe an image in the color space. The values of the pixels are expressed as an array of "opacity" value $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\alpha \in [0, 1]$, with 0 being for background and 1 being for foreground. In the GrabCut algorithm, the image is taken to consist of pixels Z_n in RGB colour space through creating multivariate GMMs (Gaussian mixture models) with K components for background and foreground regions. A vector $k = \{k_1, \ldots, k_n, \ldots, k_N\}$ is introduced, with $k_n \in \{1, \ldots, K\}$ assigned to each pixel.

The Gibbs energy of GrabCut algorithm for segmentation becomes

$$E(\alpha, k, \theta, z) = U(\alpha, k, \theta, z) + V(\alpha, z), \tag{1}$$

where E is defined as Gibbs energy, U is the data term to evaluate the fit of the opacity distribution α to the data z, V is the smoothness term, and the parameter θ describes grey-level distributions of image foreground and background and consists of histograms of grey values as equations (2)–(4):

$$\theta = \{h(z; \alpha), \quad \alpha = 0, 1\}, \tag{2}$$

$$U(\alpha, k, \theta, z) = \sum_n D(\alpha_n, k_n, \theta, z_n), \tag{3}$$

where

$$D(\alpha_n, k_n, \theta, z_n) = -\log p(z_n | \alpha_n, k_n, \theta) - \log \pi(\alpha_n, k_n), \tag{4}$$

where $p(\cdot)$ is a Gaussian probability distribution and $\pi(\cdot)$ is the mixture weighting coefficient. So,

$$D(\alpha_n, k_n, \theta, z_n) = -\log \pi \alpha_n, k_n + \frac{1}{2} \log \det \Sigma \alpha_n, k_n + \frac{1}{2} [z_n - \mu \alpha_n, k_n]^T \sum \alpha_n, k_n^{-1} [z_n - \mu \alpha_n, k_n]. \tag{5}$$

Therefore, the parameters of the model are as follows:

$$\theta = \{\pi(a, k), \mu(a, k), \Sigma(a, k), \alpha = 0, 1, k = 1, \ldots, k\}, \tag{6}$$

where the parameter θ depends on the weights π, means μ, and covariances Σ After learning the three parameters, we can get the grey-level distributions of image foreground and background. The smoothness term V is computed by using Euclidean distance as follows:

$$V = (a, z) Y \sum_{m,n \in C} [\alpha_n \neq \alpha_m] \exp - \beta \left\| z_m - z_n \right\|^2, \tag{7}$$

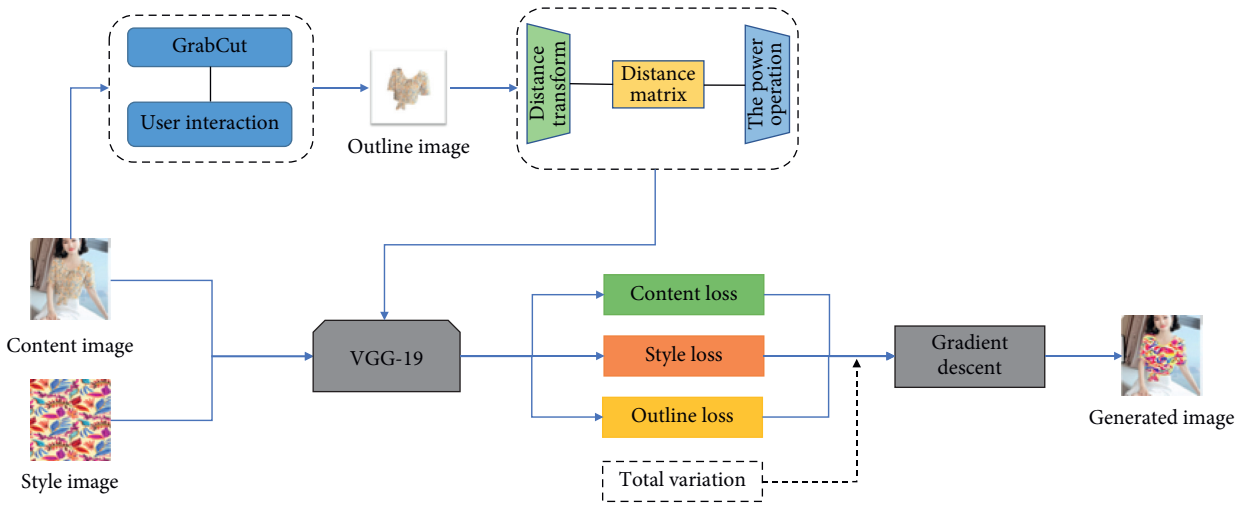

Figure 1: The framework of our proposed style transfer algorithm.

where C is the set of pairs of neighbouring pixels. This value of constant β ensures that the exponential term switches appropriately between high and low contrast.

The Gibbs energy function helps to determine whether the unknown pixels are foreground or background and also shows the difference between neighbouring pixels of an image. The algorithm finally gets the local minimum which converges at least to E. It can straightforwardly detect when E stops to decrease significantly and to terminate iteration automatically [24].

3.1.2. Results of GrabCut. Users can employ GrabCut algorithm, an interactive image segmentation method, to specifically extract any part of an image, which is in line with our purpose of extracting desired clothing from complex background to generate outline image. The results are shown in Figure 2.

Through our experiments, we find that in these four cases, the bounding rectangle alone is not sufficient to enable foreground extraction to be completed. The models' necks or arms are dragged into the rectangle, so sometimes they are mistakenly regarded as foreground. Therefore, we have to make some simple brush strokes on the clothing parts. Black means selecting areas of background, and white means selecting areas of foreground. According to Figure 2, we can see that there is no need for complicated strokes. Users simply brush the boundaries with a few strokes. Therefore, the operation is acceptable.

3.2. Style Transfer. The basic principle of neural style transfer is to extract the content features and style features of the input images and to use the pretrained convolutional neural network (CNN) to mix them to generate new images. In this paper, input images include content image with complex background, outline image that is extracted from content image by GrabCut, and style image. They are input into the convolutional neural network, and by constraining the content loss, style loss, and outline loss, a stylized clothing image G is generated, which simultaneously matches the shape of the original clothing and the style of the style image. In this paper, the trained VGG-19 network model is used as the image feature extractor [26].

3.2.1. Content and Style Loss. With a given input image to the CNN, filter responses to every layer are produced as feature map. Feature maps on some layers can be regarded as the content representation. Gatys et al. considered the feature responses in higher layers of the network as the content representation [12]. The layer l with N_l distinct filters has N_l feature maps, and M_l describes the height times the width of the feature map. Therefore, the content representation of an image in the layer l can be defined as a matrix $F^l \in R^{N_l \times M_l}$, where F^l_{ij} is the activation of the i^{th} filter at position j in layer l. Therefore, content image $\vec{p}$ and the random noise image $\vec{x}$, which is obtained by initializing content image through network, have their respective content feature representation P^l and F^l in layer l. Content loss between the two feature representations is defined as follows:

$$L_c(\vec{p}, \vec{x}, 1) = \frac{1}{2} \sum_{i,j} \left(F^l_{ij} - p^l_{ij}\right)^2. \tag{8}$$

The derivative of this loss is defined as follows:

$$\frac{\partial L_c}{\partial F^l_{ij}} = \begin{cases} \left(F^l - P^l\right)_{ij}, & \text{if } F^l_{ij} > 0, \\ 0, & \text{if } F^l_{ij} < 0. \end{cases} \tag{9}$$

To obtain style representation of an input image, Gatys et al. used a feature space. This feature space is built on top of the filter responses in any layer of the network to capture texture information. According to Gatys, textures of an image are per definition stationary, so a texture model needs to be agnostic to spatial information [12]. Thus, our style representation should discard the spatial information in the feature maps, which is given by the correlations between the

Figure 2: Outline images in the bottom row are extracted from content images in the top row by interactive GrabCut algorithm. Images in the middle row display all user interactions: black (background brush); white (foreground brush).

different filter responses [12]. These feature correlations are obtained by the Gram matrix $G^l \in R^{N_l \times M_l}$, where G^l_{ij} is the inner product between the feature maps i and j in the layer l as shown in the following equation:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}, \tag{10}$$

where k represents the k^{th} element of the feature map. A set of Gram matrices $\{G^1, G^2, \ldots, G^l\}$ from layers 1, 2, ..., l in the network in response to a given style image provides a stationary description of the texture, which fully specifies a style in our model. To generate a new style on the basis of a given image, we use a random noise image to find another image that matches the Gram-matrix representation of the original image by using gradient descent. And the optimization is done by minimizing the mean-squared distance between the entries of the Gram matrix of the generated image and the Gram matrix of the original image. Style image $\vec{a}$ and the random noise image $\vec{x}$ have their respective style feature representation A_l and G_l in layer l. The contribution of layer l to the total loss is expressed as follows:

$$E = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G^l_{ij} - A^l_{ij}\right)^2, \tag{11}$$

and the total style loss is

$$L_s(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l, \tag{12}$$

where w_l are weighting factors of the contribution of each layer to the total loss. The derivative of E_l can be computed as follows:

$$\frac{\partial E_l}{\partial F^l_{ij}} = \begin{cases} \frac{1}{N_l^2 M_l^2} \left(\left(F^l\right)^T \left(G^l - A^l\right) \right)_{ji}, & \text{if } F^l_{ij} > 0, \\ 0, & \text{if } F^l_{ij} < 0. \end{cases} \tag{13}$$

3.2.2. Outline Loss. With neural style transfer, the style from the style image is transferred to the whole content image. However, in the fashion field, in order to get a desired clothing design, the following two points must be constrained. Firstly, keep the shape of the original clothes; secondly, ensure that the style is only transferred to the desired clothes in the content image, and the rest of the areas keep clean.

For decorated logo generation, Atarsaikhan et al. proposed a new loss function based on distance transform of the input image, which allows the preservation of the silhouettes of text and objects, constraining style transfer only around

the designated area [27]. Therefore, based on this method, we introduce an outline loss function, which is obtained by converting outline image to binary image and then conducting distance transform. The concept of distance transform was first proposed by Rosenfeld and Pfaltz [28]. The basic idea is to convert a binary image into grayscale image where each object pixel has a value corresponding to the minimum distance from the background. Take binary image as an example (Figure 3), where the pixel values inside the outline are 1, and the other pixel values are 0. Distance transform calculates the distance between each pixel and the nearest outline boundary, and the pixel values inside the outline become 0.

In this paper, the outline image pixels are divided into the inside pixels p of the outline and the outside pixels q of the outline. By calculating the distance between the inside pixels and the outside pixels in Euclidean distance metric, the distance matrix D is obtained as follows:

$$Dp, q = \mathrm{Min}\left(\mathrm{disf}\left(p, q\right)\right), \tag{14}$$

$$\mathrm{disf}\left(p\left(x_1, y_1\right), q\left(x_2, y_2\right)\right) = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2}. \tag{15}$$

The farther the distance, the larger the value of the pixel, the clearer the outline, and the better the shape of clothes. With distance transform, the values of the inside pixels of outline become zero. Therefore, in order to increase the distance, we take pixelwise power of outside pixels of outline, with power of two or more [27] as follows:

$$d_{ij} = \begin{cases} 0, & \text{if inside of the outline,} \\ d_{ij}^n, & \text{otherwise.} \end{cases} \tag{16}$$

Through emphasizing the specific group of pixels, we get the distance feature. Given a random noise image $\overrightarrow{x}$ and an outline image $\overrightarrow{o}$, the distance feature is $D_{\overrightarrow{x}c}$ and $D_{\overrightarrow{o}c}$. The distance loss L_d is defined as follows:

$$L_d = \frac{1}{2}\left(D_{\overrightarrow{x}c}^n - D_{\overrightarrow{o}c}^n\right)^2. \tag{17}$$

3.2.3. Style Transfer. In this paper, style transfer is guided by the difference between the Gram-matrix representation of the content image, the style image, the outline image, and the generated image. The difference is represented by the loss function.

The loss functions used in this paper include content loss L_c for preserving the content of the content image, style loss L_s for preserving the style of the style image, and outline loss L_d for preserving the original shape of clothes and limiting the style transfer region. Suppose that α, β, and γ are the weighting factors for them, respectively; then, L_{total} is defined as follows:

$$L_{\mathrm{total}} = \alpha L_c + \beta L_s + \gamma L_d. \tag{18}$$

3.2.4. Total Variation Regularization. In order to smooth and denoise the boundary region, total variation norm R_{TV} is employed. Then, we can combine total variation norm R_{TV} and the total loss functions into total loss L_{total} by taking linear addition. In this paper, KL diversity, which is commonly used in image classification and prediction, is chosen as the loss function in the proposed method.

$$R_{\mathrm{TV}} = \frac{D_x^2}{N_{D_x}} + \frac{D_y^2}{N_{D_y}}, \tag{19}$$

where D_x and D_y, respectively, represent the transverse difference and longitudinal difference of the generated image, while N_{D_x} and N_{D_y} represent the number of elements of the corresponding difference results.

4. Experimental Results

4.1. Dataset and Experimental Set. Higher layers in the network capture the high-level content in terms of objects and their arrangement in the input image but do not constrain the exact pixel values of the reconstruction [12]. In contrast, reconstructions from the lower layers simply reproduce the exact pixel values of the original image [12]. Therefore, we use different layers to extract features. In this paper, the pretrained VGG-19 network is applied, consisting of 16 convolutional layers and 5 pooling layers. We reconstruct the content of the input images from layers conv4-2 and reconstruct the style of the input images from layers conv1-1, conv2-1, conv3-1, conv4-1, and conv5-1. Weighting factors for content image and style image, α and β, are 0.001and 0.8, respectively. γ is 1.0, and the weight of R_TV is 0.001.

Figure 4 shows different values of weighting factor γ. Outline loss L_d is designed to preserve the shape of clothing. Through experiments, we can see that with the increase of weighting factor γ, the noises from outside of the clothing shapes can be removed. Therefore, style transfer can be more tightly constrained within the outline area.

Figure 5 shows results with different emphasizing power n. When there is no emphasis ($n = 1$), there is a lot of noise around the clothes. When increasing the emphasizing power n, the noise is reduced a lot. Therefore, we can assume that the larger the emphasis on power, the more impressive the generated clothes' design. By taking pixelwise power of a matrix, the values of the outside pixels of outline become larger, and the values of the inside pixels of outline keep zero, which greatly enlarges the difference and separates the clothes and background.

The experimental pictures selected in this paper are from pop-fashion.com and Taobao.com.

4.2. Comparison with Other Methods. In Figure 6, we compare our method with other three types of style transfer methods. (1) NeuralST [12]: Gatys et al. showed artistic neural style transfer by generating a new image that combines both the content of the content image and the style of the style image. (2) MSG-Net [16]: Zhang and Dana built a multistyle generative network, which achieves real-time performance. (3) Style swap: Chen and Schmidt proposed a

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

5	4	3	2	2	3	4	5
4	3	2	1	1	2	3	4
3	2	1	0	0	1	2	3
3	2	1	0	0	0	1	2
2	1	0	0	0	0	1	2
3	2	1	0	0	1	2	3
4	3	2	1	1	2	3	4
5	4	3	2	2	3	4	5

(b)

Figure 3: (a) The input outline image and (b) the distance transform result.

(a) (b) (c)

Figure 4: The results with different values of weighting factor γ. When increasing the values of γ, the noise is removed a lot. (a) $\gamma = 0.00001$. (b) $\gamma = 0.01$. (c) $\gamma = 1.0$.

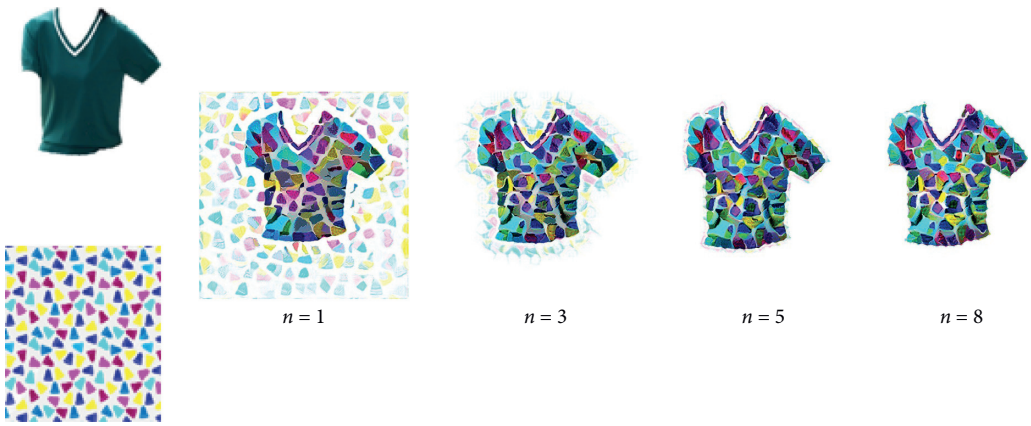

Figure 5: The results with different emphasizing power n. When there is no emphasis ($n = 1$), there is a lot of noise around the clothes. When increasing the emphasizing power n, the noise is reduced a lot.

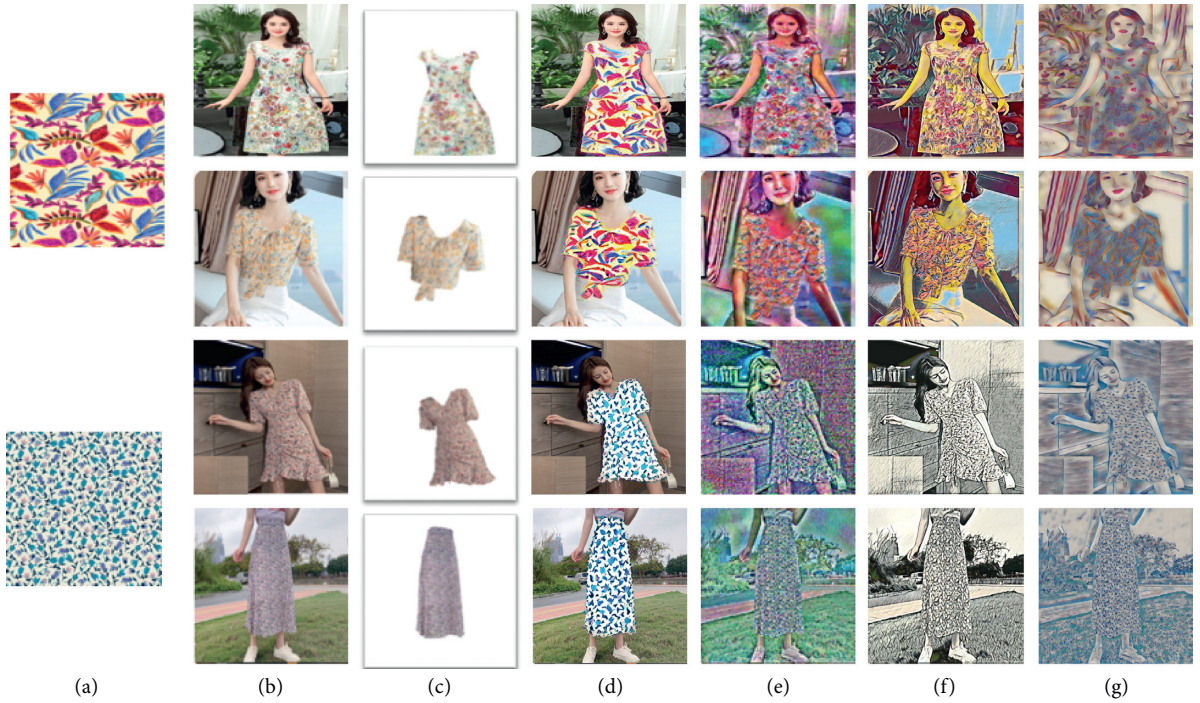

Figure 6: The first left column shows the input style images. The second left column shows four input content images. In our method, given style images, content images, and outline images of third column, new designs of clothes are generated. In other three methods, given style images and content images, new designs of clothes are generated. (a) Style image. (b) Content image. (c) Outline image. (d) Ours. (e) Gatys et al. [12]. (f) Zhang and Dana [16]. (g) Chen and Schmidt [17].

simpler optimization objective based on local matching that combines the content structure and style textures in a single layer of the pretrained network [17].

When comparing these methods, we find that only our method keeps the background clean. In other methods, the style from the style image is transferred to the whole content image rather than only clothes. Compared with Gatys, our method appears to better preserve the detailed textures of the style images and have less noise. The flowers and leaves of style images are very clear. Compared with Zhang and Dana, our method better transfers the colour of the style image. In the second last column, although the generated images preserve the textures of style images, they may lose the colour feature. In the last column, no matter the feature of content or style, they are both not well synthesized. However, in our method, the patterns of style images and the original global structure are faithfully blended.

4.3. Experimental Results. In order to show better results and make more sense, we have added more experiments and showed more examples. We picked several arts with different

Figure 7: The style transfer results with images containing big and specific patterns as style images. (a) Style image. (b) Content image. (c) Outline image. (d) Iteration 100. (e) Iteration 600. (f) Iteration 1000.

Figure 8: The results with oil paintings and watercolour paintings as style images. (a) Style image. (b) Content image. (c) Outline image. (d) Iteration 100. (e) Iteration 600. (f) Iteration 1000.

styles, like abstract arts, watercolour paintings, and oil paintings, and thus we got different effects.

Firstly, we picked two style images with big and specific patterns and picked two shirts as content images. We found it difficult to transfer such big patterns or figures from style images to clothes. However, the second line shows that the colour boundary can be identified automatically, so with different colour in original clothes, the style can be transferred variously. The results are shown in Figure 7.

Secondly, in Figure 8, we picked two oil paintings and one watercolour painting as style images. These paintings do not have specific patterns, so these new images also do not generate any new ones. Fortunately, we can see that the style of painting, including colour, has been transferred to clothes perfectly and becomes a new and beautiful design.

Thirdly, the three style images are random Internet photos. We can see that the colour has been transferred to clothes perfectly, and the new patterns will be randomly generated depending on the original ones in style images. Figure 9 shows the results.

Lastly, three style images with various stripes are used in Figure 10, and the results show that both the shape and the colour of stripes can be transferred perfectly to clothes. We conclude that the more specific and smaller the

(a) (b) (c) (d) (e) (f)

Figure 9: The results with random Internet pictures as style images. (a) Style image. (b) Content image. (c) Outline image. (d) Iteration 100. (e) Iteration 600. (f) Iteration 1000.

(a) (b) (c) (d) (e) (f)

Figure 10: The results with images containing stripes as style images. (a) Style image. (b) Content image. (c) Outline image. (d) Iteration 100. (e) Iteration 600. (f) Iteration 1000.

patterns are in style images, the more perfect they can be transferred.

5. Conclusion

This paper proposed an interactive image localized style transfer method especially for clothes. Through combining users' favourite clothing images and style images, a new design of clothes can be generated, which allowed ordinary users to easily design their own clothes and also provided inspiration for professional designers. In order to keep the shape of the original clothes and focus on localized transfer, we introduced a third image called outline image, which was extracted from content image by interactive GrabCut algorithm. Finally, the experiment results showed that the images generated by the method in this paper

outperformed other methods. The approach in this paper has simple operations, high efficiency, and certain practicability.

Given the simplicity of our method, we believe that there is still substantial room for improvement. In future works, we plan to explore more advanced algorithm to allow more kinds of pictures, including both realistic and artistic images, as style inputs. More diverse image datasets will be used for training and testing to make the generated clothing styles more fashionable and more impressive.

Data Availability

All data included in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Beijing Municipal Natural Science Foundation (no. 4172014), Project of High-Level Teachers in Beijing Municipal Universities in the Period of 13th Five-Year Plan (no. CIT&TCD201804031), and R&D Program of Beijing Municipal Education Commission (no. KM202010011011).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS 2012)*, pp. 1097–1105, Red Hook, NY, USA, December 2012.

[2] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Parsing clothing in fashion photographs," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3570–3577, Providence, RI, USA, June 2012.

[3] Y. Kalantidis, L. Kennedy, and L. Li, "Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos," in *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, pp. 105–112, New York, NY, USA, April 2013.

[4] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1096–1104, Las Vegas, NV, USA, June 2016.

[5] Y. Ma, J. Jia, S. Zhou et al., "Towards better understanding the clothing fashion styles: a multimodal deep learning approach," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 38–44, San Francisco, CA, USA, February 2017.

[6] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis,," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 229–238, New York, NY, USA, September 1995.

[7] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, 2000.

[8] B. Julesz, "Visual pattern discrimination," *IEEE Transactions on Information Theory*, vol. 8, no. 2, pp. 84–92, 1962.

[9] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, Washington, DC, USA, September 1999.

[10] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of the SIGGRAPH*, vol. 34, New Orleans, LA, USA, July 2000.

[11] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: a review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3365–3385, 2020.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks,," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, Las Vegas, NV, USA, June 2016.

[13] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, October 2016.

[14] Y. H. Li, N. Y. Wang, J. Y. Liu, and X. D. Hou, "Demystifying neural style transfer," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2230–2236, Melbourne, Australia, August 2017.

[15] D. G. Chen, L. Yuan, J. Liao, N. H. Yu, and G. Hua, "StyleBank: an explicit representation for neural image style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2770–2779, Honolulu, HI, USA, July 2017.

[16] H. Zhang and K. Dana, "Multi-style generative network for real-time transfer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 349–365, Munich, Germany, September 2018.

[17] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," in *Proceedings of the NIPS Workshop Constructive Machine Learning*, pp. 1–9, Barcelona, Spain, December 2016.

[18] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proceedings of the European Conference on Computer Vision*, pp. 702–716, Amsterdam, The Netherlands, October 2016.

[19] C. Li and M. Wand, "Combining Markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2479–2486, Las Vegas, NV, USA, June 2016.

[20] H. Xiong, J. Wu, Q. Liu, and Y. Cai, "Research on abnormal object detection in specific region based on Mask R-CNN," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, 2020.

[21] H. Xiong, Q. Liu, S. Shao, and Y. Cai, "Region-based convolutional neural network using group sparse regularization for image sentiment classification," *EURASIP Journal on Image and Video Processing*, vol. 2019, p. 30, 2019.

[22] B. Wang, H. Xiong, and C. Jiang, "A multicriteria decision making approach based on fuzzy theory and credibility mechanism for logistics center location selection," *The Scientific World Journal*, vol. 2014, Article ID 347619, 9 pages, 2014.

[23] S. Jiang and Y. Fu, "Fashion style generator," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial*

Intelligence, pp. 3721–3727, Melbourne, Australia, August 2017.

[24] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": interactive foreground extraction using iterated graph cuts," in *Proceedings of the ACM SIGGRAPH 2004*, pp. 309–314, Association for Computing Machinery, Los Angeles, CA, USA, August 2004.

[25] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proceedings of the Eighth IEEE International Conference on Computer Vision ICCV 2001*, vol. 1, pp. 105–112, Vancouver, Canada, July 2001.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference On Learning Representations*, San Diego, CA, USA, April 2015.

[27] G. Atarsaikhan, B. K. Iwana, and S. Uchida, "Contained neural style transfer for decorated logo generation," in *Proceedings of the 2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pp. 317–322, Vienna, Austria, April 2018.

[28] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *Journal of the ACM*, vol. 13, no. 4, pp. 471–494, 1966.