02-13

16th International Conference on Machine Vision Applications (MVA)
National Olympics Memorial Youth Center, Tokyo, Japan, May 27-31, 2019.

Face Style Transfer and Removal with Generative Adversarial Network

Qiang Zhu
Simon Fraser University
Vancouver, BC, Canada
qza64@sfu.ca

Ze-Nian Li
Simon Fraser University
Vancouver, BC, Canada
li@sfu.ca

Abstract

In this paper, we present a method to transfer the style of a stylized face to another face without style and recover photo-realistic face from the same stylized face image simultaneously. Here style refers to the local patterns or textures of some existing paintings. Style transfer gives a new way for artistic creation while style removal can be beneficial for face verification or photo-realistic content editing. Our approach contains two components: the Style Transfer Network (STN) and the Style Removal Network (SRN). STN renders the style of the stylized image to the non-stylized image and the SRN is designed to remove the style of a stylized photo. By applying the two networks successively to an original input photo, the output should match the input photo. The experiment results in a variety of portraits and styles demonstrate our approach's effectiveness.

1 Introduction

Style transfer plays a vital role in image manipulation and creates new artistic works in different artistic styles from existing photographs. The inverse problem of recovering photo-realistic faces from corresponding artistic portraits is also investigated in this paper. In both tasks, we need to preserve the identity. In the style transfer task, reconstructing images from deep features may lead to extreme distortions which can result in identity loss. Generally, stylized faces can have diverse facial expressions and the facial details are distorted, so it is not easy to recover photo-realistic face images from stylized faces. Thus, both tasks are challenging.

A variety of methods have been proposed for style transfer. Previous CNN based methods leverage a pre-trained network to extract deep features and match their gram matrix statics to recombine the style of artistic work and the content of a given photo [3, 2, 7, 9, 12, 17, 18, 19].

Style transfer and removal can also be posed as a domain adaption problem [21, 8, 16, 1]. Inspired by these methods which can transfer an image from a source image domain to have images appearance similarity with images in a target domain, we introduce a way to transfer style to a face photo, where the style is from an example face of another person. At the same time, we can also remove the style of the stylized face. We use two asymmetric networks (Figure 1): one to transfer style and another one to remove style. The style transfer network takes a source face image and a stylized face image as input, while the style removal network only requires the stylized face image as input. Both transform networks should preserve the identity of the source face image. We utilize the style removal network to help maintain the identity in the style transfer process. And we also employ the identity-preserving and the pixel-level Euclidean loss functions to constrain the recovered faces to lie on the manifold of faces without style while preserving its identity. Finally, we leverage adversarial loss to ensure that we can obtain satisfying visual results.

To sum up, our main contributions are:

- A transform network that can transfer the style of a stylized face photo to a source face photo and recover photo-realistic face from the same stylized face image.
- We unite an adversarial loss, a cycle-consistent loss, an identity-preserving loss and a pixel-level similarity loss to transfer style and recover face.

2 Related Work

2.1 Style Transfer and Style Removal

Given a style image and a content image, the traditional style transfer methods render the style of the style image to the content image to produce a new stylized image. Most recent style transfer approaches utilize Convolutional Neural Networks (CNNs), but use different loss function and diverse optimization method [4, 2, 9, 10, 17]. Gatys *et al.* [4] first proposed to employ a CNN for style transfer which is to use optimization to match the gram matrix correlation statistics. Johnson *et al.* [9] speed up the process by training a feed-forward network using perceptual loss functions.

Different from style transfer, style removal is to digitally remove the style of the stylized image [1, 16]. In our work, we simultaneously perform both tasks and we demonstrate that better results can be gained by improving the processes of transfer and removal in turn.

2.2 Generative adversarial networks

Style transfer and style removal can also be treated as a domain adaptation problem which is to learn a mapping between the source image domain and target image domain. Many researchers have used generative adversarial networks (GAN) [5] for the mappings of two image domains and have achieved appealing results in style transfer [11], image editing [20] and image generation [15]. Isola *et al.* introduced a "pix2pix" framework which learns a mapping between the input and the output images utilizing a conditional generative adversarial network [6]. To learn the mapping, Zhu *et al.* proposed CycleGAN which adopted generative network with a cycle consistency loss to make the distribution of the mapped images cannot be distinguished from that of real images in the target domain. Based on the CycleGAN model, Chang *et al.* [1] proposed Paired-CycleGAN to transfer an arbitrary makeup style to another photo. In our work, we employ cycle architecture together with variants of cycle-consistency loss to transfer the style of the reference stylized face to the source face and remove the style of the stylized face at the same time.

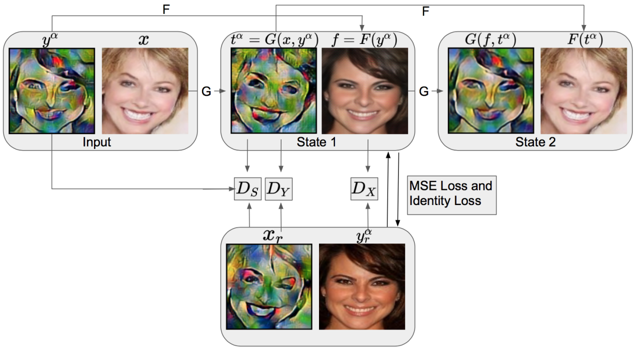

Figure 1. The architecture of our framework contains two parts: a style transfer network G and a style removal network F which are learned simultaneously. G learns to render the style of y^α to x while F learns to remove the style of y^α. Adversarial discriminators D_Y aims to distinguish between the real stylized faces x_r and samples generated by G, and vice versa for F and D_X. While D_S is used to determine if two pairs of faces are stylized with the same style. The results in the first state are used as input to generate images in the second state. Then by comparing the output of the second state with the input, we aim to preserve identity and style consistency.

3 Method

Let X and Y be the no-style and with-style image domains and we have training samples $\{x_i\}_{i=1}^N$ where $x_i \in X$ and $\{y_j\}_{j=1}^M$ where $y_j \in Y$. $Y^\alpha \in Y$ stands for a sub-domain of Y that consists of images of a particular style α. We denote y_j as y_j^α while $y_j \in Y^\alpha$. Denote data distribution of X and Y as $x \sim p_X$ and $y \sim p_Y$.

As illustrated in Figure 1, our framework consists of two networks: $G : X \times Y^\alpha \rightarrow Y^\alpha$ and $F : Y \rightarrow X$. We train the two networks G and F at the same time, G is designed to render a particular style and F is used to remove style. We feed network G with an image of a face with style, $y^\alpha \in Y^\alpha$, and a picture of a different face without style, $x \in X$. Style transfer network G learns to extract the style of y^α and renders it to x while preserving the identity of x. While the style removal network F learns to remove the style of the same photo y^α maintaining its identity. Note that if G and F work successfully, we can transfer the style of the output of G to the output of F which will double the number of training samples. And if G and F can maintain identity, we can attain two images that look like the two input images. Thus, based on the above analysis, we have the following losses.

Adversarial loss. We utilize an adversarial loss to encourage the results of G to be indistinguishable from the real stylized samples from domain Y, the loss is defined as:

$$\begin{aligned} L_G(G, D_Y) =& \mathbb{E}_{y^\alpha \sim p_Y}[logD_Y(x_r)] \\ &+ \mathbb{E}_{x \sim p_X, y^\alpha \sim p_Y}[log(1 - D_Y(G(x, y^\alpha)))] \end{aligned}$$

Where G is encouraged to generate faces $G(x, y^\alpha)$ indistinguishable from the real samples , while D_Y aims to distinguish between the reference stylized faces x_r from domain Y and the translated faces $G(x, y^\alpha)$. We also introduce a similar adversarial loss to force F to generate images that look similar to the non-stylized reference faces from domain X:

$$\begin{aligned} L_F(F, D_X) =& \mathbb{E}_{x \sim p_X}[logD_X(y_r^\alpha)] \\ &+ \mathbb{E}_{y^\alpha \sim p_Y}[log(1 - D_X(F(y^\alpha))] \end{aligned}$$

A variant of Cycle Consistent loss. We argue that the learned mapping functions should have cycle-consistence property. For every image y^α in domain Y and x in domain X, our image generation network should be capable of bringing x and y back to the input image. This is to say, if we transfer style to x and then remove the style immediately, we could obtain the image x exactly. And if we stylize face x with the style of face y^α, and then transfer the same style of the result $G(x, y^\alpha)$ back to the style-removed face $F(y^\alpha)$, the result $G(F(y), G(x, y^\alpha))$ should look similar to the input face y^α. So the cycle consistent loss function is defined as:

$$\begin{aligned} L_{cyc}(G, F) =& \mathbb{E}_{x \sim p_X, y^\alpha \sim p_Y} \|F(G(x, y^\alpha)) - x\|_1 \\ &+ \mathbb{E}_{x \sim p_X, y^\alpha \sim p_Y} \|G(F(y^\alpha), G(x, y^\alpha)) - y^\alpha\|_1 \end{aligned}$$

Style loss. Inspired by [8], we also employ an assisting discriminator D_S to determine if two pairs of faces are stylized with the same style. When we train the model, we need to feed D_S with two style pairs, one is fake style pairs (y^α, $G(x, y^\alpha)$) and another one is real style pairs (y, the same style α rendered to a different face).

$$L_S(G, D_S) = \mathbb{E}_{x\sim p_X, y^\alpha\sim p_Y}[log D_S(y^\alpha, x_r)] + \mathbb{E}_{x\sim p_X, y^\alpha\sim p_Y}[log(1 - D_S(y^\alpha, G(x, y^\alpha)))]$$

Where x_r is a synthetic ground-truth generated by the current style transfer algorithm [13].

MSE loss. We enforce the style-removed face $F(y^\alpha)$ to be indistinguishable from its ground-truth y_r^α. The pixel-wise L_1 loss function between $F(y^\alpha)$ and y_r^α is as follows;

$$L_{MSE}(F) = \|F(y^\alpha) - y_r^\alpha\|^2$$

Identity loss for F. To maintain the identity of the style-removed faces, we encourage the style-removed face $F(y^\alpha)$ and the ground-truth face y_r^α to have similar feature representations which are computed by VGG-19 pre-trained network. The identity-preserving loss L_{id} is expressed as:

$$L_{id} = \mathbb{E}_{(y, y_r^\alpha)\sim p(y, y_r^\alpha)} \|\phi(F(y^\alpha)) - \phi(y_r^\alpha)\|^2$$

Here $\phi(.)$ denotes the activations of the layer ReLU3-2 of the VGG-19 [22] pre-trained network while processing some input image.

Total loss. The loss L is defined as:

$$L = \lambda_G L_G + \lambda_F L_F + L_{cyc} + \lambda_S L_S + L_{MSE} + L_{id}$$

λ_G, λ_F, λ_S are the hyperparameters. And we set $\lambda_G = \lambda_F = \lambda_S = 0.2$.

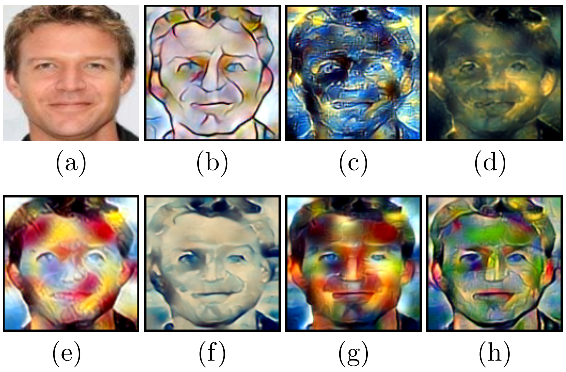

Figure 2. Samples of the synthesized dataset. (a) Original real face image. (b)-(h) The stylized faces of (a) from Mosaic, Starry-night, the shipwreck of the minotaur, Candy, Wave, Rain-princess,and La-muse.

4 Experiments

4.1 Datasets

To train our network, we need four separate datasets, two containing faces without style and another two containing faces with a wide variety of styles. We utilize the CelebA [14] dataset to generate such datasets. Firstly, we randomly select 1K source face from the dataset and then resize them to get 128×128×3 RGB images. These images are used as real ground-truth faces y_r^α. To generate stylized faces, we use the universal style transfer [13] method for 200 diverse styles. And finally we harvest 5K training pairs for $\{y^\alpha, y_r^\alpha\}$ pairs. Then using another 1K real face and the same 500 styles, we obtain 5K training pairs for $\{x_r, x\}$ pairs in the same way. To test our network, we utilize 500 real faces to generate 2K testing pairs for $\{y^\alpha, y_r^\alpha\}$ pairs from ten various styles and use another 500 real faces to generate 2K testing pairs for $\{x_r, x\}$ pairs. As shown in Figure 2 , we applied different styles to a single source face producing vivid stylized results. There is no overlap between the training and testing datasets.

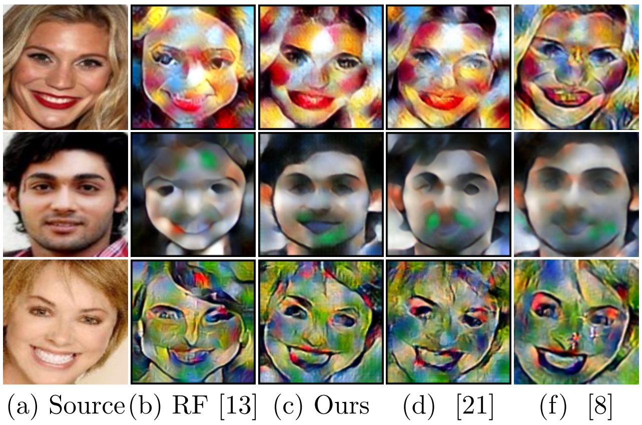

Figure 3. Results of style transfer. We compare with style transfer work [21, 8].

4.2 Style Transfer Results

Figure 3 shows the style transfer results. Our network is able to transfer a wide variety of artistic styles across diverse source faces preserving the identity of them. We also compare our method with two different previous work [21, 8].

CycleGAN [21] is an unsupervised approach that uses unpaired dataset for the image-to-image translation task. It utilizes generative networks to make the mapped images and the real samples in the target domain have the same data distribution. To use it for style transfer, we employed a collection of stylized faces with the same style and a collection of source faces to train the network. The learned mapping function takes one source face as input and transforms it into the specific stylized face domain. CycleGAN can only transfer images between two specific domains and for each style, it needs a set of faces with that style to train a network. Thus, it is not appropriate for our style transfer task. As shown in Figure 3(d), the mouth, nose, and eyes of the stylized faces are distorted and preservation of facial identity is lost.

We also compare our work with [8], a conditional generative adversarial network, called pix2pix. Since it can also only transfer images between two specific domains, it is not suitable for our task. For style transfer task, we use a paired stylized faces and source faces to train the network. The resolution of the stylized output faces of the network is 256 × 256 pixels, and we resize them to 128 × 128 pixels for comparison. From the fifth column of Figure 3, one can see that the stylized faces are fuzzy and distorted. Thus, their network fails to generate attractive results.

Compared with previous work, we obtain more appealing results and perform better in preserving the facial identity of the source face photo, as shown in Figure 3(c).

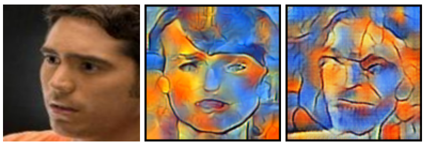
(a) Source (b) Reference (c) our

Figure 4. Limitations of STN.

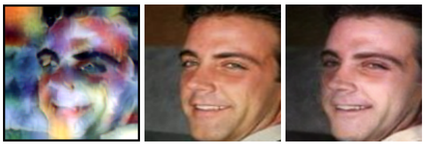
(a) Stylized (b) GT (c) Our

Figure 5. Limitations of SRN.

4.3 Style Removal Results

In Figure 6, we compare our results with three different methods by using our training dataset to retrain the three approaches.

We retrain CycleGAN [21] using a set of stylized faces and a set of source faces. The network learns a mapping between two different domains. The learned mapping function takes one stylized face as input and transforms it into the real face domain. Since CycleGAN uses unpaired face datasets, it fails to map the features of the source faces to the stylized faces. As a result, their approach is not appropriate for style removal task. As shown in Figure 6(c), the recovered faces are distorted and the style that overlaps with the hairs was not fully removed.

We compare our work against [8]. Since it employs a patch-based convolutional neural network to discriminate the image patch between a source face and a stylized face, their method fails to catch the global structure. For style removal task, we use a paired stylized faces and real faces to train the network. From the fourth column of Figure 6, one can see that the details of the face were fuzzy and it loses identity consistency with respect to the source face.

Fatemeh and Xin [16] propose a generative network for style removal. They employ a full connected layer in the generator to match feature maps between the source faces and the stylized faces. However, we find that the residual network is more suitable for this task. This is because the de-stylized output faces should be identity consistent with real faces. we don't require our layer to learn how to produce a new image taking a source image as input. Instead, we just require it to learn how to generate an output by adjusting the input image. Instead of directly learning a desired underlying mapping, a residual layer just needs to learn a residual mapping, so it is more appropriate for this task. As seen in Figure 6(e), while the method can produce acceptable results, the recovered faces lose some details of hairs and color consistency.

Compared with the above approaches, the results of our method demonstrate better preservation of face identity and are more consistent with the source faces in colors, as shown in Figure 6(f).

5 Limitations

However, one limitation of our method is that the network may result in artifacts for faces that have large pose variations. Also, since the color information of the stylized portraits is distorted, it is not easy to recover the color corresponding to that of the real images. As shown in Figure 4 and Figure 5, the style transfer result has some distortions and the recovered face loses some details of face colors.

6 Conclusions

We present a method for transferring style from a reference face to a source face and for removing style of the same stylized face. We train the style transfer network and removal network together, which allows them to strength each other. Our network can extract the facial features of the reference face and apply it to a source face. At the same time, it can also de-stylize stylized portraits successfully.

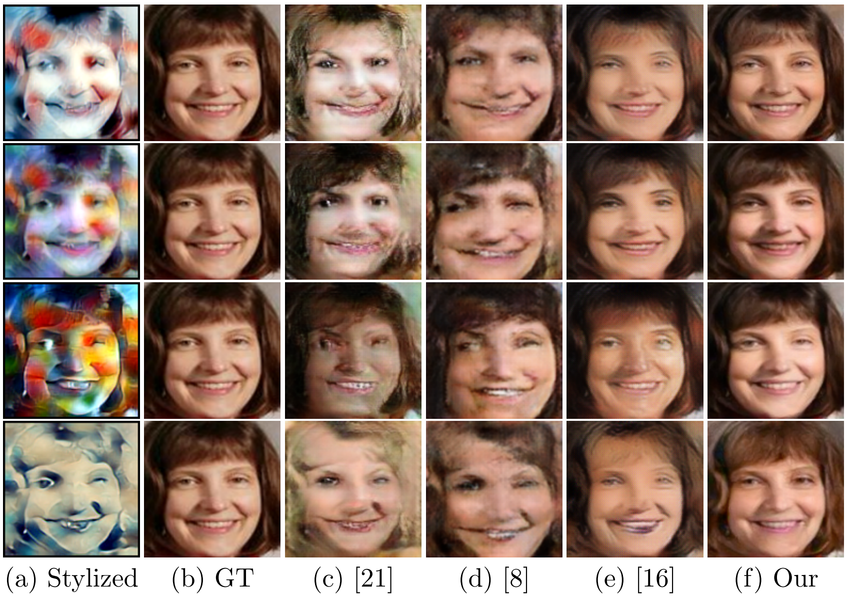

Figure 6. Results of style removal. We compare with style removal work [21, 8, 16].

References

[1] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. 2016.

[3] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *Proc. of ICLR*, 2017.

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, 2016.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, 2016.

[7] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[9] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[10] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016.

[11] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[12] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified texture synthesis with feed-forward networks. In *Proc. CVPR*, 2017.

[13] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pages 386–396, 2017.

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[15] Rómer Rosales, Kannan Achan, and Brendan J Frey. Unsupervised image translation. In *iccv*, pages 472–478, 2003.

[16] Fatemeh Shiri, Xin Yu, Piotr Koniusz, and Fatih Porikli. Face destylization. In *Digital Image Computing: Techniques and Applications (DICTA), 2017 International Conference on*, pages 1–8. IEEE, 2017.

[17] D Ulyanov, A Vedaldi, and VS Lempitsky. Instance normalization: the missing ingredient for fast stylization. corr abs/1607.0 (2016).

[18] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and

Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016.

[19] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. 2017.

[20] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017.

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