Face Destylization

Fatemeh Shiri, Xin Yu, Piotr Koniusz, Fatih Porikli

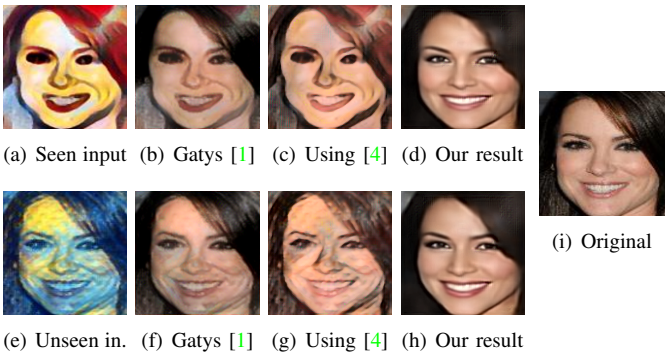
***Abstract*—Numerous style transfer methods which produce artistic styles of portraits have been proposed to date. However, the inverse problem of converting the stylized portraits back into realistic faces is yet to be investigated thoroughly. Reverting an artistic portrait to its original photo-realistic face image has potential to facilitate human perception and identity analysis. In this paper, we propose a novel Face Destylization Neural Network (FDNN) to restore the latent photo-realistic faces from the stylized ones. We develop a Style Removal Network composed of convolutional, fully-connected and deconvolutional layers. The convolutional layers are designed to extract facial components from stylized face images. Consecutively, the fully-connected layer transfers the extracted feature maps of stylized images into the corresponding feature maps of real faces while the deconvolutional layers are used to generate real faces from the transferred feature maps. To enforce the destylized faces to be similar to authentic face images, we employ a discriminative network, which consists of convolutional and fully connected layers. We demonstrate the effectiveness of our network by conducting experiments on an extensive set of synthetic images. Furthermore, we illustrate our network can recover faces from stylized portraits and real paintings for which the stylized data was unavailable during the training phase.**

I. Introduction

Applying artistic styles to existing photographs has attracted much attention in both academia and industry with several interesting applications. The inverse problem of reverting an artistic portrait back to its photo-realistic version is investigated in this paper. Revealing the latent real faces can provide essential information for human perception, computer analysis and photo-realistic multimedia content editing. Since facial details and expressions in stylized portraits often undergo severe distortions and become contaminated with artifacts such as profile edges and color changes e.g., as in Fig. 1(a) and Fig. 1(e), recovering a photo-realistic image of face from its stylized version is very challenging.

The seminal work of [1] stylizes the content of an arbitrary image according to a given reference artwork and achieves appealing style transfer results, hovewer, its iterative optimization procedure is computationally costly. Several methods based on feed-forward neural networks [2]–[9] accelerate the style transfer for specific styles.

For our inverse problem, the above style transfer methods fail to recover authentic face images as shown in Fig. 1(f) and Fig. 1(g). These approaches typically use Gram matrices to capture style-related contents. Since Gram matrices are designed to measure the correlations between feature maps of a style image and a target face, the spatial structure of an output image is not guaranteed to be similar to the target face. Therefore, existing style transfer methods which rely on Gram matrices are not sufficient for restoring photo-realistic portraits.



(a) Seen input (b) Gatys [1] (c) Using [4] (d) Our result (i) Original

(e) Unseen in. (f) Gatys [1] (g) Using [4] (h) Our result

Fig. 1. Comparison to the state-of-art methods. (a) and (e) 128×128 stylized face images in *Candy* style (which is seen and used for training) and in *Starry Night* style (which is unseen style), respectively. (b, f) Results obtained by applying [1] for the given stylized faces. (c, g) Results obtained by applying [4]. (d, h) Our destylization results. (i) 128×128 ground-truth face image (used for evaluation purposes; not available to the algorithm for training).

To capture local statistics of a style image, some approaches use a so-called patch-based Generative Adversarial Network (GAN) [10], [11]. However, patch-based GANs do not take the global structure of faces into account thus a direct application of patch-GAN may not produce satisfactory results. We will show later that patch-based methods [10], [11] fail to attain consistency of face colors. For the inverse problem, the patch-based GAN methods result in even bigger inconsistencies.

We note that the state-of-the-art style transfer methods [2], [4], [10] do not fully take into consideration how to extract facial features from different stylized images and then recover realistic images of faces. Our goal is to reveal the latent real face images from multiple style portraits (seen styles) and achieve destylization even when the styles are not available in the training dataset (unseen styles).

To this end, we propose a novel destylization network that automatically maps the stylized faces to photo-realistic ones in an end-to-end fashion. Our network is composed of two components: a generative part, named *Style Removal Network (SRN)*, and a discriminative part. SRN constitutes convolutional, fully-connected and deconvolutional layers. The convolutional layers are exploited to extract facial components from stylized face images. As we aim to generate realistic face images, a fully-connected layer is developed to map the extracted feature maps of stylized faces to the feature maps of real faces. Then the mapped feature maps are projected to the image domain, thus forming face images. The discriminative network enforces the latent space of SRN to produce realistic images of faces, in the manner similar to [12]–[14]. We train the entire network on a large-scale dataset of stylized and

real face pairs. Our proposed framework can restore important facial details and attributes thanks to the style removal and discriminative subnetworks.

Furthermore, we observe that the filters of Convolutional Neural Network (CNN) learned during training (seen styles) are able to extract features from images containing unseen styles. Thus, the facial information of stylized portraits can be extracted and used to represent features of real faces. Moreover, our network can also restore the images of faces given an unseen style. In the experimental section, we demonstrate that our network is able to recover realistic faces from both seen and unseen styles e.g., synthesized and original portraits and paintings.

Below, we summarize our main contributions:

- We propose FDNN which is able to generate photo-realistic faces from stylized ones. The results resemble accurately to the ground-truth images of faces in terms of facial properties e.g., facial profiles and expressions. To the best of our knowledge, our method is the first attempt to provide a unified framework for face destylization which can remove both seen and unseen styles (observed cf. unobserved styles during training).
- We develop a style removal sub-network to extract features from stylized input images of faces, then map these style features to real facial features and re-project them to the image domain for the purpose of generating authentic looking faces.
- We provide a dataset of pairs of the stylized and real images of faces used in our experiments to stimulate further research in destylization.

II. Related Work

We briefly review the deep generative image models, deep style transfer, and image translation approaches.

A. Deep Generative Image Models

Recently, several frameworks have been proposed for image generation, such as variational auto-encoders [15], auto-regressive models [16], and GANs [12]. Among these models, GANs generate impressive results because they employ adversarial losses that force the generated images to be indistinguishable from their real counterparts. In order to improve the stability of the training procedure of GANs, various methods have been proposed [11], [13], [17]–[20]. GANs are also employed by the style transfer [10] and cross-domain image generation [21]–[25] approaches. Li and Wand [10] train a Markovian GAN for image style transfer such that a discriminative training is applied on Markovian neural patches to capture local style statistics. However, patch-based methods may fail to capture the global structure of objects.

B. Deep Style Transfer

Style transfer methods transfer the style of a specific artwork into a given photograph. They can be divided into two categories: *image optimization-based* and *feed forward* methods.

The optimization-based method [1] transfers the style by updating pixels of the image iteratively. It minimizes the distance between Gram matrices generated from feature maps of the style and synthesized image with respect to input noise. The approach [26] initializes the optimization algorithm with a content image instead of noise. Li and Wand [27] use Markov Random Field (MRF) in the deep feature space to enforce local patterns. The work [28] employs linear models to transfer styles and to preserve colors by matching color histograms. Gatys *et al.* [29] detect and control spatial, color and scale factors during the stylization process. Moreover, [30] proposes a multi-modal CNN to perform stylization hierarchically with multiple losses formed across multiple scales. In [31], the loss function is improved by imposing a histogram-based loss. The above optimization-based methods require a time-consuming iterative optimization process, which limits their practical application.

In contrast, *feed-forward* approaches replace the original on-line iterative optimization procedure by off-line training to produce stylized images through a single forward pass [2], [4], [10]. Johonson *et al.* [4] train the generative network by perceptual loss functions. The architecture of their generator network follows work [32]. However, they additionally use residual blocks and replace pooling layers by so-called fractionally strided convolutions. In a concurrent work, [2] use a multi-resolution architecture for their generator network. Li and Wand [10] pre-compute a Markovian GAN which captures the feature statistics of patches. To achieve faster convergence, Ulyanov *et al.* [3], [33] replace batch with instance normalization in the generator. These feed-forward approaches [2]–[4], [10] are three orders of magnitude faster than optimization-based style transfer methods. However, these networks only transfer images according to a predefined style and thus they need to be re-trained for every new style. Some recent approaches improve the style transfer from a single style to multiple styles [5], [7]. Dumoulin *et al.* [5] propose to train a style transfer network for multiple styles by the use of conditional instance normalization. Given feature activations of the content and style images, [7] replaces the content features with the closest-matching style features patch-by-patch. A recent summary of state-of-the-art stylization methods can be found in [34].

C. Image Transformation

Mapping images from one domain to another has a wide range of applications. The idea of image transformation comes from so-called image analogies [35] which focus on the non-parametric patch-based texture synthesis from a single input-output training image pair. Methods [11], [13], [14], [19], [32], [36], [37] employ neural networks to learn a parametric translating function from a large dataset of input-output pairs, such as super-resolution and colorization. Isola *et al.* [11] propose the pix2pix framework to learn a mapping from input to output by a conditional GAN. Similar ideas have been applied to generating photographs from sketches [36], semantic layout and scene attributes [37].

Moreover, [11] also uses a convolutional patchGAN classifier for its discriminator network. The above patch-based method does not take the global structure of faces into account. Furthermore, their network employs the architecture "Unet" to transfer the source to the target domain and utilizes low-level features in the generative part that can result in distorted facial images. In contrast, our approach takes into account the global structure of faces and learns how to extract usuful features for face destylization.

III. Method

Our FDNN network has two components: (i) a Style Removal Network (SRN), which transforms stylized faces to the photo-realistic ones, and (ii) a discriminative network, which enforces the generated faces by SRN to be indistinguishable from the real faces. Figure 2 illustrates the overall architecture of our proposed network.

A. Style Removal Network

In Fig. 2, our SRN is enclosed by green frame. SRN aims at removing various styles of portraits and generating realistic faces. Our SRN comprizes convolutional layers followed by batch normalization layers, a fully connected layer and deconvolutional layers followed by batch normalization layers. The convolutional layers are employed to extract facial features from stylized face images. Then, we incorporate a fully-connected layer to transfer the extracted feature maps of stylized images into the feature maps of real faces. In order to synthesize images of real faces, deconvolutional layers project these transferred feature maps to the image domain.

In order to train SRN, we use stylized portraits as inputs and their corresponding ground-truth images of real faces as desired supervising output signals. Since a dataset of portrait/real face pairs is not readily available, we opt to generate a large number of stylized faces in numerous styles from real face images. Figure 3(c) and Fig. 3(f) illustrate the effectiveness of SRN.

B. Discriminative Network

Using only Euclidean distance, i.e. ℓ_2 loss, between the destylized faces and the corresponding ground-truth real ones tends to generate over-smoothed results as shown in Fig. 3(c) and Fig. 3(f), and this phenomenon is also mentioned in [14]. Therefore, a class-specific discriminative objective is also incorporated into our SRN, aiming to enforce the destylized face images to lie on the same latent space of the authentic face images.

As shown in the red frame of Fig. 2, the discriminative network is constructed by convolutional layers and fully connected layers. It is employed to determine whether an image is sampled from real face images or the destylized ones. With the help of the discriminative loss, also known as adversarial loss, we can generate destylized faces more similar to real ones. In doing so, the adversarial loss is back-propagated to update the parameters of SRN. Figure 3(d) and Fig. 3(g) illustrate the impact of the adversarial loss on the final results.

C. Training Details

Our FDNN is trained in an end-to-end manner. We use Stylized Face (SF) and Real Face (RF) ground-truth image pairs (s_i, r_i) as our training dataset, where r_i represents the real face images aligned by eyes only, and s_i is a synthesized SF image from r_i. For each real face r_i, we generate eight different SFs i.e., Edvard Munch's *Scream*, *Candy*, *Feathers*, *Starry Night* by Van Gogh, *la Muse* by Pablo Picasso, Wassily Kandinsky's *Composition VII*, *Mosaic* and Francis Picabia's *Udnie*, and obtain SF/RF training pairs. The stylized faces of *Scream*, *Candy* and *Feathers* are used in the training stage. As detailed in Sec. IV, we find that these distinct portraits provide a sufficient training data for our needs.

Our training strategy enforces the generated face $\hat{r}_i$ to be similar to its corresponding ground-truth r_i. Therefore, we employ a pixel-wise ℓ_2 loss between $\hat{r}_i$ and r_i, and we minimize the objective $Q(\mathcal{T})$ of SRN as follows:

$$\begin{aligned}\min_{\mathcal{T}} \quad Q(\mathcal{T}) &= \mathbb{E}_{(\hat{r}_i, r_i)\sim p(\hat{r}, r)} \|\hat{r}_i - r_i\|_F^2 \\ &= \mathbb{E}_{(s_i, r_i)\sim p(s, r)} \|G_{\mathcal{T}}(s_i) - r_i\|_F^2,\end{aligned} \tag{1}$$

where $\mathcal{T}$ indicates the parameters of SRN generator G, $p(s, r)$ represents the joint distribution of the SF and RF images in the training dataset and $p(\hat{r}, r)$ represents the joint distribution of destylized and the ground-truth faces.

To achieve high-quality results, we force SRN to fool the discriminative supervising network that employs a binary classifier which task is to distinguish whether incoming image samples contain real or generated faces. Similar to the idea of [12], [32], [38], our goal is to make the discriminative network fail to distinguish generated faces from real ones. Hereby, we maximize the adversarial loss of the discriminative network $F(\mathcal{L})$ as follows:

$$\begin{aligned}\max_{\mathcal{L}} \quad F(\mathcal{L}) &= \mathbb{E}\left[\log D_{\mathcal{L}}(r_i) + \log(1 - D_{\mathcal{L}}(\hat{r}_i))\right] \\ &= \mathbb{E}_{r_i\sim p(r)}[\log D_{\mathcal{L}}(r_i)] + \mathbb{E}_{\hat{r}_i\sim p(\hat{r}))}[\log(1 - D_{\mathcal{L}}(\hat{r}_i))],\end{aligned} \tag{2}$$

where $\mathcal{L}$ represents the parameters of the discriminative network D, $p(r)$ and $p(\hat{r})$ indicate the distributions corresponding to the real and the generated faces, respectively, and $D_{\mathcal{L}}(r_i)$ and $D_{\mathcal{L}}(\hat{r}_i)$ are the outputs of D. Since the loss F is back-propagated to update not only the parameters $\mathcal{L}$ but also $\mathcal{T}$, we also minimize the objective function $Q_f(\mathcal{T})$ of SRN:

$$\begin{aligned}\min_{\mathcal{T}} \quad Q_f(\mathcal{T}) &= \mathbb{E}_{(s_i, r_i)\sim p(s, r)} \|G_{\mathcal{T}}(s_i) - r_i\|_F^2 \\ &\quad + \lambda \mathbb{E}_{s_i\sim p(s))}[\log(1 - D_{\mathcal{L}}(G_{\mathcal{T}}(s_i)))],\end{aligned} \tag{3}$$

where scalar λ is a trade-off between supervising the generator by the ground-truth data vs. the discriminator supervision, respectively.

Since each layer in our FDNN is differentiable, we employ the Root Mean Square Propagation (RMSprop) [39] to update

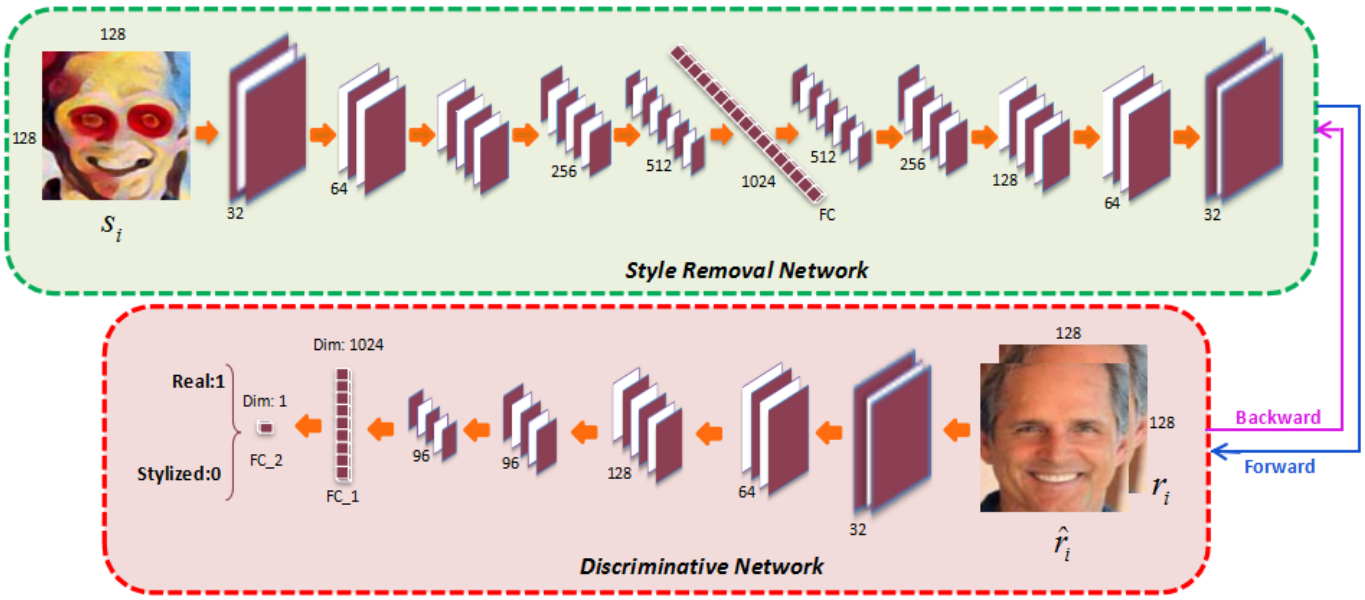


Fig. 2. Face destylization neural network consists of two parts: a generative network (green frame) and a discriminative network (red frame).

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

Fig. 3. Contribution of each component in FDNN. (a) Ground-truth real face images. (b) Input portrait of *Feathers* from training styles and (e) input portrait of *la Muse* from unseen styles (from test dataset; not available to the algorithm for training). (c, f) Destylization results without adversarial loss. (d, g) Our final results.

$\mathcal{T}$ and $\mathcal{L}$. In order to maximize the adversarial loss F, the stochastic gradient ascent is used to update $\mathcal{L}$:

$$\begin{aligned} \Delta^{i+1} &= \beta\Delta^i + (1-\beta)(\frac{\partial F}{\partial \mathcal{L}})^2, \\ \mathcal{L}^{i+1} &= \mathcal{L}^i + \alpha\frac{\partial F}{\partial \mathcal{L}}\frac{1}{\sqrt{\Delta^{i+1}+\epsilon}}, \end{aligned} \tag{4}$$

where α and β represent the learning and the decay rate respectively, i is the iteration index, Δ is an auxiliary variable, and ϵ is set to 10^{-8} to avoid division by zero. For SRN, both losses Q and F are used to update $\mathcal{T}$ by the stochastic gradient descent:

$$\begin{aligned} \Delta^{i+1} &= \beta\Delta^i + (1-\beta)(\frac{\partial Q_f}{\partial \mathcal{T}})^2, \\ \mathcal{T}^{i+1} &= \mathcal{T}^i - \alpha(\frac{\partial Q_f}{\partial \mathcal{T}})\frac{1}{\sqrt{\Delta^{i+1}+\epsilon}}, \end{aligned} \tag{5}$$

We set $\lambda = 0.01$ to limit supervision of the generator by the discriminator and allow appearance-based learning from the ground-truth image pairs. As the iterations progress, the output faces will resemble the real faces more. Therefore, we gradually reduce the impact of the discriminative network by decreasing λ,

$$\lambda^n = \max\{\lambda \cdot 0.995^n, \lambda/2\}, \tag{6}$$

where n is the index of the epochs. Eqn. 6 not only increases the impact of the appearance similarity term but also preserves the class-specific discriminative information in the training phase.

D. Implementation Details

Similar to [12], [32], we employ batch normalization after the convolutional and deconvolutional layers of SRN except for the last deconvolutional layers. We also use leaky rectified linear units (leakyReLU) with a negative slope 0.2 as non-linear activation functions. For training, the learning rate α is set to 0.001 and multiplied by 0.99 after each epoch, and the decay rate is set to 0.01. The discriminative network is only

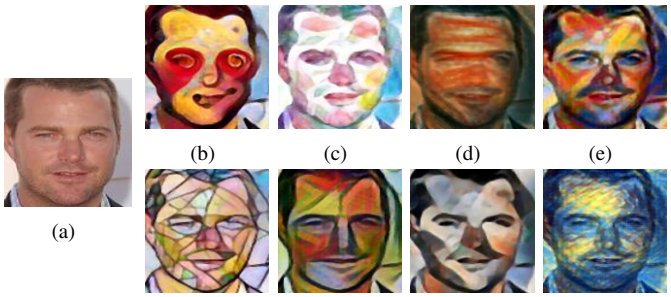


Fig. 4. Illustration of the synthesized dataset. (a) Original real face image. (b)-(d) The synthesized stylized faces of (a) form *Candy, Feathers* and *Scream* which have been used for training our network. (e)-(i) The synthesized stylized faces of (a) form *Composition VII, Mosaic, la Muse, Udnie* and *Starry* styles which have not been used for training.

employed in the training phase. In the testing phase, we feed a stylized face image into the SRN to obtain its realistic version.

IV. Synthesized Dataset

Training of a deep neural network requires a large number of samples to prevent models from overfitting to the training data. The publicly available large-scale face datasets [40], [41] only provide faces in the wild but not pairs of real images of faces and their stylizations. Therefore, we opt to generate a large number of stylized faces from the corresponding real face images in eight distinct styles: *Starry Night, la Muse, Composition VII, Scream, Candy, Feathers, Mosaic* and *Udnie*. To generate such a dataset, there are a number of alternative feed-forward approaches available [2]–[4]. We choose the recent feed-forward style transfer model [4].

We firstly select at random 10K images of cropped real faces (within $\pm 30^{\circ}$ orientation) from the CelebA [41] dataset for training and 1K images for testing, and then resize them to 128×128 pixels. We use 10K training images as our real ground-truth faces r_i. To generate three different portraits of each face, we retrain the style transfer model [4] for *Scream, Candy* and *Feathers* styles separately. Finally, we obtain 30K SF/RF pairs for training our network. We also use 1K test real faces to generate 8K SF/RF face pairs from eight different styles (each test face corresponds to eight distinct styles) for testing our network. Figure 4 shows the stylized samples that are generated from a single real image containing a face (Fig. 4(a)).

V. Experiments

We compare our method qualitatively and quantitatively against four different state-of-the-art methods. As explained in Sec. IV, we gather 30K SF/RF face pairs from three styles as a training dataset and 8K SF/RF pairs faces generated from different eight styles for testing. In all the cases, the ground-truth real faces and the corresponding stylized faces do not overlap in the training and testing datasets. Our method is feed-forward and works real-time.

A. Qualitative Evaluation

Comparison to the state of the art. Firstly, we note that the test stylized face images are not used by us during the training of our model. The resolution of stylized and destylized output faces in this study is 128×128 pixels. We compare our approach against four various approaches as detailed below.

We compare our work against [1] which is an image-optimization based style transfer method that has not any training stage. To generate real faces, this network strives to preserve the contents of a portrait and the style of the corresponding photo-realistic face. The network fails to produce appealing results as illustrated in Fig. 5(c) and Fig. 6(c). Because of how the Gram matrix is constructed, this method only captures the correlation between feature maps of style and synthesized images. Thus, the spatial arrangement at the pixel level is not preserved.

We also use a feed-forward approach [4] for destylization. Due to Gram matrix, this method also produces distorted facial details. As shown in the first row of Fig. 5(d), the edges of the face have been blurred and the color of the face is not consistent. From the first row of Fig. 6(d), one can see that the style overlapping with the eyes has not been fully removed. Thus, their network fails to restore authentic looking eyes.

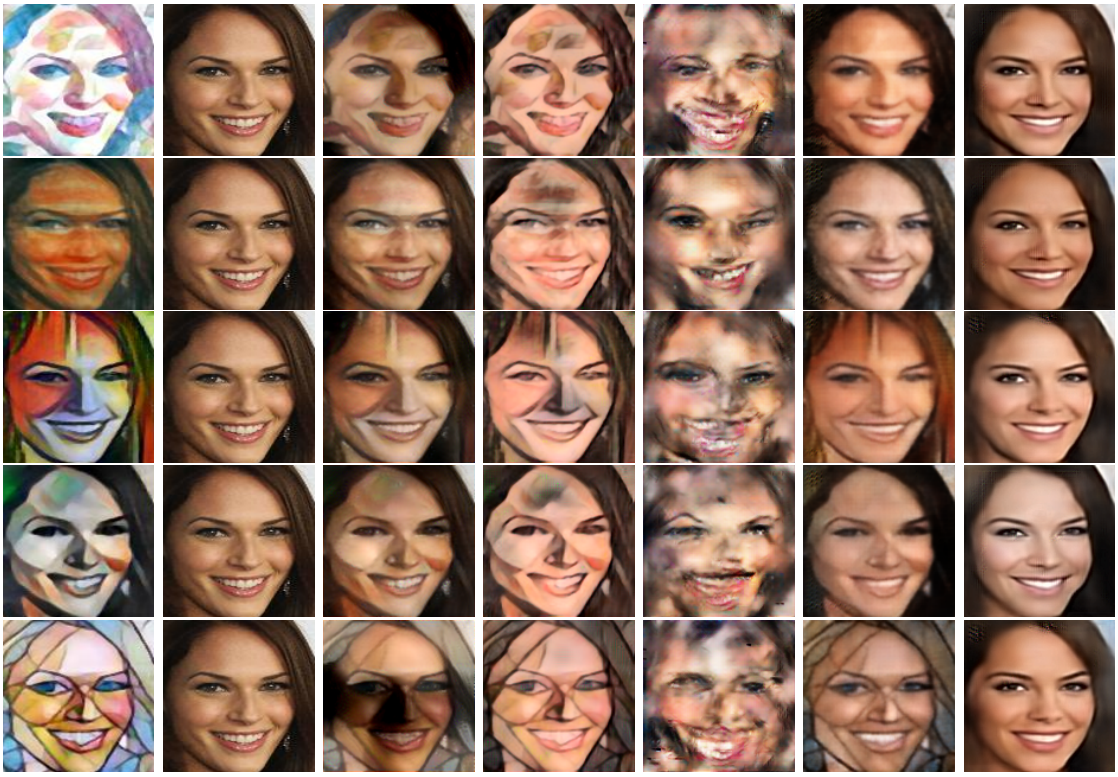
Li and Wand [10] propose a patch-based style transfer method, known as Markovian GAN. We use their network for destylization and apply their standard protocols. As such a method is trained with stylized face patches, it cannot capture the global structure of facial images. As seen in Fig. 5(e) and Fig. 6(e), the facial color consistency cannot be preserved either. In contrast, our method produces highly-consistent facial colors and captures the global structure of faces well.

Isola *et al.* [11] present a general image-to-image translation method, known as pix2pix. It employs the architecture "Unet" for the generator network and uses a convolutional patch based neural network as the discriminator network. The discriminator network is trained to classify whether an image patch represents a sample of real or generated face. In addition, the low-level features from the bottom layers of Unet also participate in generating images of faces. These low-level features corrupt the destylized images and are the cause of poor removal of styles in the images e.g., for unseen styles. As shown in Fig. 5(f) and Fig. 6(f), while pix2pix can produce acceptable results for seen styles, it fails to remove styles effectively from unseen style. As shown in the fourth row of Fig. 6(f), obvious artifacts appear in the generated face of an unseen style.

Our destylized results demonstrate higher fidelity w.r.t. the real faces, better consistency in colors and even preserve the identity of the subject, as shown in Fig. 5(g) and Fig. 6(g).

B. Quantitative Evaluation

Face Reconstruction. In Tab. I, we report the reconstruction performance measured on the entire test dataset for each approach. We use the average Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) [42] scores for which higher scores indicate better results.



(a) SF (b) Groundtruth (c) Gatys [1] (d) Johnson [4] (e) MGAN [10] (f) pix2pix [11] (g) Ours

Fig. 5. Results of the state-of-the-art methods for face destylization. (a) Input portraits of *Feathers, Scream* from seen styles as well as *la Muse* , *Udnie* and *Mosaic* from unseen styles (from test dataset; not available to the algorithm during training) (b) Ground-truth images of real faces.

TABLE I
COMPARISON OF PHYSICAL (PSNR) AND PERCEPTUAL (SSIM) QUALITY MEASURES FOR THE ENTIRE TEST DATASET.

Method	Seen Styles		Unseen Styles	
	PSNR	SSIM	PSNR	SSIM
Gatys [1]	22.6792	0.8656	20.2320	0.8493
Johnson [4]	22.8481	0.8745	21.2184	0.8632
MGAN [10]	19.5254	0.8548	17.2645	0.8270
pix2pix [11]	22.9893	0.8871	21.6316	0.8860
Ours	**23.2086**	**0.9087**	**22.4430**	**0.9015**

TABLE II
COMPARISON OF CONSISTENCY BETWEEN DESTYLIZED FACES FROM VARIOUS SEEN AND UNSEEN STYLES.

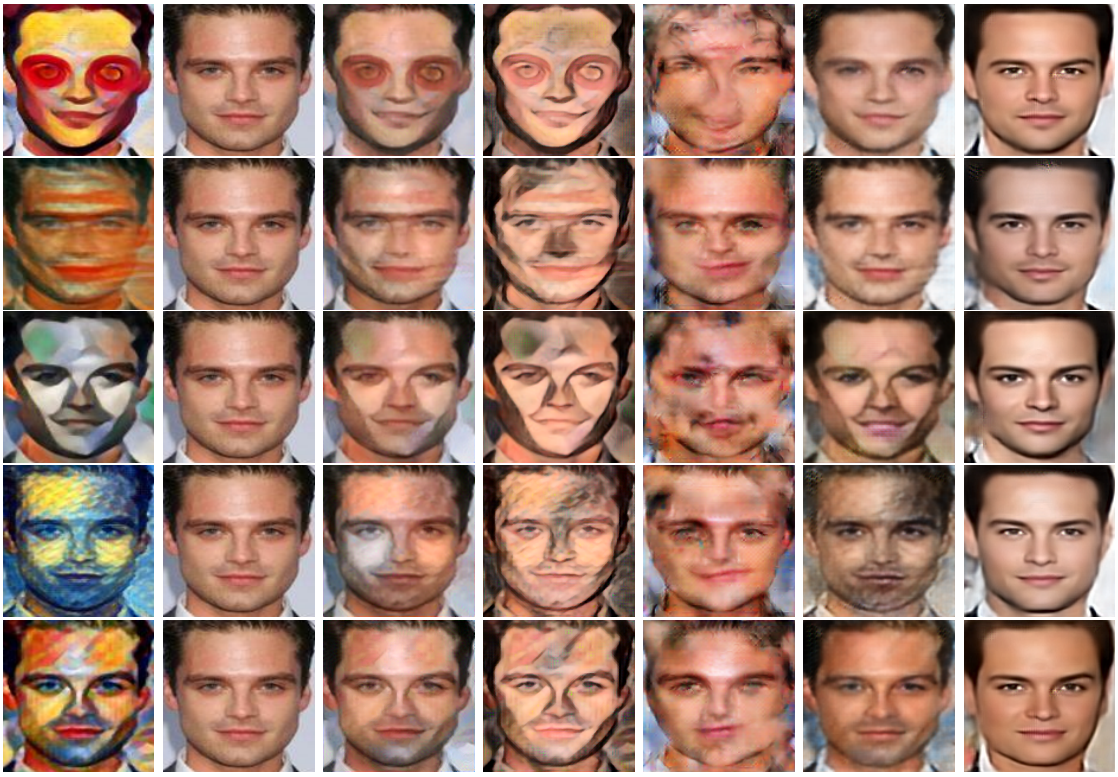
	Seen Styles	Unseen Styles
Gatys [1]	82%	83%
Johnson [4]	73%	72.5%
MGAN [10]	2%	1%
pix2pix [11]	93.33%	85.1%
Ours	**98%**	**90.8%**

We report performance of destylization algorithms for two scenarios: seen and unseen styles. For the seen styles, results of the state-of-the-art style transfer methods are shown in the first and second rows of Fig. 5 and Fig. 6. For the destylization of portraits of unseen styles, we demonstrate results in the third, fourth and fifth rows of Fig. 5 and Fig. 6.

Tab. I shows that our results achieve better PSNR and SSIM than the state-of-the-art methods on seen styles and unseen styles. This performance also coincides with the visual results.

Consistency Analysis. Intuitively, the destylized faces from the different styles of the same person should look similar. Examples generated from multiple styles are shown in Fig. 5(g) and Fig. 6(g). In this experiment, we demonstrate that our method not only recovers realistic faces with high fidelity but also generates faces looking close to each other given multiple styles of the same person on input. This indicates that SRN can indeed extract facial features from portraits despite different styles and transfer these features to recover underlying faces.

To evaluate the consistency of generated faces from different portraits of the same person, we adapt the off-the-shelf deep face recognition approach [43]. First, we randomly choose 100 RF and 800 corresponding SF faces from eight different styles in the test dataset for our gallery (three seen styles and five unseen styles). Then, we employ Gatys [1], Johnson [4], MGAN [10], pix2pix [11] and our FDNN to recover real faces from eight various stylized faces. For each method, we set 100 destylized faces from the *Candy* style as a query dataset and set the other 700 destylized faces from the other seven styles as a search dataset. Following the standard protocol, we



(a) SF (b) Groundtruth (c) Gatys [1] (d) Johnson [4] (e) MGAN [10] (f) pix2pix [11] (g) Ours

Fig. 6. Result of the state-of-the-art methods for face destylization. (a) Input portraits of *Candy* and *Scream* from seen styles as well as *la Muse, starry Night* and *Mosaic* from unseen styles (from test dataset; not available to the algorithm during training) (b) Ground-truth images of real faces.

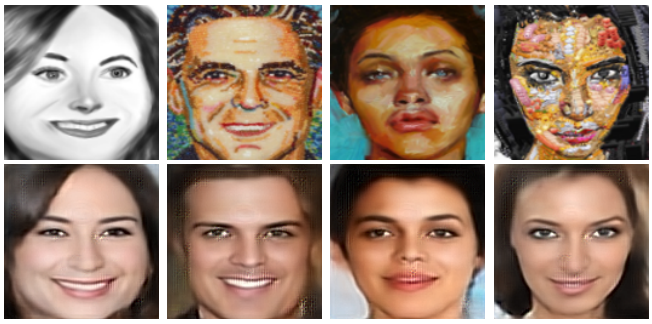
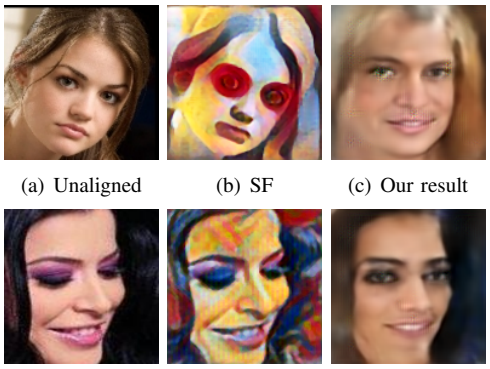


Fig. 7. Results for the original paintings. Top row: the original portraits from DevianArt. Bottom row: our destylization results.



(a) Unaligned (b) SF (c) Our result

(d) Upright pose (e) SF (f) Our result

Fig. 8. Failures. (a) An unaligned ground-truth face. (e) Stylized face of (a). (c) Our result. (d) An upright pose. (e) Stylized face of (d). (c) Our result.

compute the Face Recognition Rate (FRR) which quantifies if the correct person is retrieved within the top-5 candidates (the probability of successful retrieval by chance is 0.71%). We also use the same procedure for other styles. Table II shows the average FRR of each method for seen and unseen styles. Our method yields high consistency score for both seen and unseen styles. This indicates the effectiveness of our FDNN in producing realistic faces of high-fidelity.

C. Performance on Original Paintings

Despite our method is trained on a synthetic dataset, it can efficiently generalize to real paintings/portraits. To demostrate this, we randomly choose some paintings with faces from DevianArt. We crop images of these faces and then align them to the CelebA face dataset in an off-line pre-processing step. Our method successfully reconstructs plausible facial details from real paintings as shown in Fig. 7. This highlights that our method is not restricted to synthesized stylized faces.

D. Limitations

Our proposed network requires that the eyes of stylized faces to be aligned beforehand to a template. Without such an alignment, FDNN may cause artifacts. However, we plan to automatically align the stylized facial images in our future work. As illustrated in Fig. 8(a), destylization is performed

on an unaligned stylized face. As a consequence, our network cannot localize facial features correctly and produces erroneous feature maps. In addition, our method may produce artifacts for portraits suffering from large pose variations, such as profile views of faces etc. Since there are not enough images of faces in side-view in the training dataset, this produces artifacts. As shown in Fig. 8(f), the network fails to generate satisfying results for an upright pose. Exploring how to address large pose variations will be our future work.

VI. Conclusion

We presented a face destylization method that extracts features of a stylized portrait and then exploits them to generate its corresponding photo-realistic face. Our network learns a mapping from stylized facial feature maps to realistic facial feature maps. Our network can successfully extract facial features from different styles and thus is able to destylize unseen style portraits as well.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423. 1, 2, 5, 6, 7

[2] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," *arXiv preprint arXiv:1603.03417*, 2016. 1, 2, 5

[3] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016. 1, 2, 5

[4] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *arXiv preprint arXiv:1603.08155*, 2016. 1, 2, 5, 6, 7

[5] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," 2016. 1, 2

[6] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Diversified texture synthesis with feed-forward networks," *arXiv preprint arXiv:1703.01664*, 2017. 1

[7] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," *arXiv preprint arXiv:1612.04337*, 2016. 1, 2

[8] H. Zhang and K. Dana, "Multi-style generative network for real-time transfer," *arXiv preprint arXiv:1703.06953*, 2017. 1

[9] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *arXiv preprint arXiv:1703.06868*, 2017. 1

[10] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," *arXiv preprint arXiv:1604.04382*, 2016. 1, 2, 5, 6, 7

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint arXiv:1611.07004*, 2016. 1, 2, 3, 5, 6, 7

[12] I. Goodfellow, J. Pouget-Abadie, and M. Mirza, "Generative Adversarial Networks," in *NIPS*, 2014. 1, 2, 3, 4

[13] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494. 1, 2

[14] X. Yu and F. Porikli, "Ultra-resolving face images by discriminative generative networks," in *ECCV*, 2016. 1, 2, 3

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 2

[16] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016. 2

[17] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," *arXiv preprint arXiv:1612.04357*, 2016. 2

[18] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016. 2

[19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242. 2

[20] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017. 2

[21] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," *arXiv preprint arXiv:1612.05424*, 2016. 2

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456. 2

[23] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477. 2

[24] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *arXiv preprint arXiv:1703.00848*, 2017. 2

[25] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *arXiv preprint arXiv:1703.05192*, 2017. 2

[26] R. Yin, "Content aware neural style transfer," *arXiv preprint arXiv:1601.04568*, 2016. 2

[27] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2479–2486. 2

[28] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman, "Preserving color in neural artistic style transfer," *arXiv preprint arXiv:1606.05897*, 2016. 2

[29] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," *arXiv preprint arXiv:1611.07865*, 2016. 2

[30] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," *arXiv preprint arXiv:1612.01895*, 2016. 2

[31] P. Wilmot, E. Risser, and C. Barnes, "Stable and controllable neural texture synthesis and style transfer using histogram losses," *arXiv preprint arXiv:1701.08893*, 2017. 2

[32] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. 2, 3, 4

[33] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," *arXiv preprint arXiv:1701.02096*, 2017. 2

[34] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song, "Neural style transfer: A review," *arXiv preprint arXiv:1705.04058*, 2017. 2

[35] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 327–340. 2

[36] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," *arXiv preprint arXiv:1612.00835*, 2016. 2

[37] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," *arXiv preprint arXiv:1612.00215*, 2016. 2

[38] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," in *NIPS*, 2015. 3

[39] G. Hinton, "Neural Networks for Machine Learning Lecture 6a: Overview of mini-batch gradient descent Reminder: The error surface for a linear neuron." 3

[40] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, 2007. 5

[41] X. W. Ziwei Liu, Ping Luo and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015. 5

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 5

[43] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition." in *BMVC*, 2015. 6