Neural Style Transfer: A Review

Yongcheng Jing† Yezhou Yang‡ Zunlei Feng† Jingwen Ye† Mingli Song†*

† Microsoft Visual Perception Laboratory, College of Computer Science and Technology, Zhejiang University

‡ School of Computing, Informatics, and Decision Systems Engineering, Arizona State University

ycjing@zju.edu.cn yz.yang@asu.edu {zunleifeng, yejingwen, brooksong}@zju.edu.cn

Abstract

The recent work of Gatys et al. demonstrated the power of Convolutional Neural Networks (CNN) in creating artistic fantastic imagery by separating and recombing the image content and style. This process of using CNN to migrate the semantic content of one image to different styles is referred to as Neural Style Transfer. Since then, Neural Style Transfer has become a trending topic both in academic literature and industrial applications. It is receiving increasing attention from computer vision researchers and several methods are proposed to either improve or extend the original neural algorithm proposed by Gatys et al. However, there is no comprehensive survey presenting and summarizing recent Neural Style Transfer literature. This review aims to provide an overview of the current progress towards Neural Style Transfer, as well as discussing its various applications and open problems for future research.

1. Introduction

Painting is a popular form of art. For hundreds of years, people have been attracted by the art of painting with the advent of many fantastic artworks, *e.g.*, Vincent van Gogh's "The Starry Night". In the past, re-drawing an image in a particular style manually required a well-trained artist and lots of time.

Since the mid-1990s, the art theories behind the fantastic artworks have been attracting the attention of not only the artists but many computer science researchers. There are plenty of studies exploring how to automatically turn images into synthetic artworks such that everyone can be an artist. Among these studies, the advances in *Non-photorealistic Rendering* (NPR) [18, 42, 39] are inspiring and nowadays it is a firmly established field. However, the NPR algorithms are usually highly dependent on specific artistic styles (*e.g.*, oil paintings, animations) they simulate [39, 16] and cannot be easily extended to produce stylized results for other artistic styles.

Recently, inspired by the power of *Convolutional Neural Network (CNN)*, Gatys *et al*. first studied how to use CNN to reproduce famous painting styles on natural images. They obtained the image representations derived from CNN and found that the representations of image content and style were separable. Based on this finding, Gatys *et al*. [14] proposed a *Neural Style Transfer* algorithm to recombine the content of a given photograph and the style of well-known artworks. Their proposed algorithm successfully produces fantastic stylized images with the appearance of a given artwork. Figure 1 shows an example of transferring the style of the Chinese painting "Dwelling in the Fuchun Mountains" onto a photo of The Great Wall. The key idea behind this algorithm is to start from random noise as the initial result and then change the values of pixels iteratively until the desired statistical feature distribution is satisfied. Since this Neural Style Transfer algorithm does not have any explicit restrictions for the type of style images, it breaks previous approaches' constraints. The work of Gatys *et al*. opened up a new field called Neural Style Transfer, which is the process of using *Convolutional Neural Network* to migrate the semantic content of one image to different styles.

The pioneering work of Gatys *et al*. has attracted wide attentions both academically and industrially. In academia, lots of follow-up studies were proposed to either improve or extend this innovative algorithm and before long, these technologies were applied to many successful industrial applications (*e.g.*, *Prisma* [36], *Ostagram* [2], *Deep Forger* [6]). However, to the best of our knowledge, there are no comprehensive survey summarizing and discussing recent

*The corresponding author

(a) Content (b) Style (c) Content + Style

Figure 1. Example of using the Neural Style Transfer algorithm of Gatys *et al*. to transfer the style of Chinese painting (b) onto The Great Wall photograph (a). The painting that served as style is named "Dwelling in the Fuchun Mountains" by Gongwang Huang.

significant advances and challenges within this new field of Neural Style Transfer.

In this paper, we will give an overview of recent development in Neural Style Transfer. Our contributions are threefold. First, we investigate, classify and summarize recent advances in the field of Neural Style Transfer. Second, we present popular evaluation methods to compare the outputs of different Neural Style Transfer methods. Third, we introduce current and potential commercial applications in this field.

The rest of this paper is organized as follows. Section 2 categorizes existing Neural Style Transfer methods and explains these methods in detail. Some improvement strategies as well as extensions for discussed methods will be given in Section 3 and Section 4. Then Section 5 provides methodologies for evaluating stylized output of Neural Style Transfer methods. Section 6 discusses commercial applications of these Neural Style Transfer methods, including current successful usages as well as potential applications. Finally, Section 7 summarizes current challenges as well as corresponding possible solutions in this field and Section 8 concludes the paper and delineates several promising directions for future research. A list of mentioned papers in this review, corresponding codes and pre-trained models are publicly available at: `https://github.com/ycjing/Neural-Style-Transfer-Papers`.

2. A Taxonomy of Neural Style Transfer Methods

In this section, we provide a categorization of Neural Style Transfer methods. Current Neural Style Transfer methods fit into one of two categories, namely *Descriptive Neural Methods Based On Image Iteration* and *Generative Neural Methods Based On Model Iteration*. The first category transfers the style by directly updating pixels in the image iteratively, while the second category first optimizes a generative model iteratively and produces the styled image through a single forward pass.

2.1. Descriptive Neural Methods Based On Image Iteration

The Descriptive Neural Method is the first proposed neural method to transfer the style between two images. The idea is to update pixels in the (yet unknown) stylized image iteratively through backpropagation, which starts from random noise. The objective of image iteration is to minimize the total loss such that the stylized image simultaneously matches the content representation of the content image and the style representation of the style image.

One of the keys to Neural Style Transfer is the representation of style, *i.e*., the pre-defined style loss function. The style loss function is optimized to match the feature statistics of the style image. Depending on the different adopted style loss functions, Descriptive Neural Methods can be further divided into methods based on *Maximum Mean Discrepancy (MMD)* and methods based on *Markov Random Fields (MRF)*. For simplicity, we call them *MMD-based* and *MRF-based* methods.

2.1.1 MMD-based Descriptive Neural Methods

Maximum Mean Discrepancy (MMD) is a popular metric of discrepancy between two distributions, based on the mean of features in the Hilbert space [20]. Recently Li *et al*. demonstrate that style transfer can be considered as a distribution alignment process from the content image to the style image [30]. Therefore MMD can be used to measure the style discrepancy. MMD-based Descriptive Neural Methods refer to the neural methods that use MMD with different kernels as the style loss for optimization.

Many Neural Style Transfer algorithms use Gram matrices to represent the style and define the distance between the entries of the Gram matrices from the style image and (yet unknown) stylized image as the style loss function. Here we need to clarify that theoretically, matching the Gram matrices is actually equivalent to minimize the MMD with the second order polynomial kernel, which has been proved in [30]. Thus, these gram-based descriptive algorithms all be-long to the category of MMD-based Neural Methods.

The algorithm of Gatys *et al*. [14] is the first proposed MMD-based descriptive method. It is based on the image reconstruction strategy and texture synthesis algorithm. Image reconstruction is the process of using only image representations to reconstruct the image itself, which is actually an inverting process for representations of the image [33]. Therefore, the reconstructed result retains only information contained in the representations. By reconstructing representations from intermediate layers in VGG network, Gatys *et al*. observe that deep convolutional neural network is capable of extracting semantic image content from an arbitrary photograph and some appearance information from the well-known artwork. According to this observation, they build the content component of newly stylized image by penalizing the difference of high-level representations derived from stylized image and content image, and further style component by matching feature statistics of stylized image and style image. After this process they successfully obtain stylized images with high perceptual quality. The algorithm of Gatys *et al*. is actually a combination of image reconstruction strategy and texture synthesis algorithm, *i.e*., the overall procedure of iterating newly stylized images using gradient descent is similar to that of image reconstruction algorithm while the style matching approach is inspired by texture synthesis method.

Given a content image $\mathbf{x_c}$ and a style image $\mathbf{x_s}$, the algorithm in [14] tries to find a stylized image $\mathbf{x}$ that minimizes the objective:

$$\mathbf{x} = \arg\min_{\mathbf{x}} E_{total} = \arg\min_{\mathbf{x}} \alpha E_c(\mathbf{x}, \mathbf{x_c}) + \beta E_s(\mathbf{x}, \mathbf{x_s}), \tag{1}$$

where the loss E_c compares content image representation of some layer to that of the (yet unknown) stylized image, and E_s compares the entries of the Gram matrices from the style image to that of the (yet unknown) stylized image to make the stylized image match feature statistics of the style image. α and β are weighting factor for content and style reconstruction. Different from image reconstruction algorithm in [33], the objective function does not contain image prior terms as the features extracted from lower layers are already considered as the image prior.

The content loss E_c is defined by the squared Euclidean distance between feature representations of content image $\mathbf{x_c}$ in layer l and that of the (yet unknown) stylized image $\mathbf{x}$:

$$E_c\left(\mathbf{x}, \mathbf{x_c}\right) = \frac{1}{2}\sum_{i,j} ||\Phi_{ij}^l(\mathbf{x}) - \Phi_{ij}^l(\mathbf{x_c})||^2, \tag{2}$$

where Φ_{ij}^l is the representation of the i^{th} filter at position j in layer l.

For the style loss E_s, [14] uses Gram matrix G_{ij}^l to obtain correlations between filter responses. The feature space built by these correlations is considered as a representation of the style. It is originally designed to capture texture information in a texture synthesis algorithm [15]. Gram matrix G_{ij}^l is defined as the inner product between representation of i^{th} and j^{th} filter in layer l:

$$G_{ij} = \sum_k \Phi_{ik}\Phi_{jk}. \tag{3}$$

Given a style representation G_{ij}^l of the style image x_s in layer l and A_{ij}^l of the (yet unknown) stylized image, the loss in layer l can be defined as the squared Euclidean distance between these style representations:

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G_{ij}^l - A_{ij}^l\right)^2, \tag{4}$$

where N_l is the number of filters and M_l represents the size of the feature map in layer l, *i.e*., the height times the width of the feature map. Li *et al*. prove that Equation (4) is equivalent to minimizing Maximum Mean Discrepancy (MMD) with the second order polynomial kernel. The total style loss is

$$E_s(x, x_s) = \sum_{l=0}^{L} w_l E_l, \tag{5}$$

where L is the total number of layers and w_l are weighting factors of each layer, which can be tuned manually.

Eventually, the objective function in [14] can be summarized as

$$\begin{aligned} E_{total} =& \frac{\alpha}{2}\sum_{i,j} ||\Phi_{ij}^l(\mathbf{x}) - \Phi_{ij}^l(\mathbf{x_c})||^2 \\ &+ \frac{\beta}{4N_l^2 M_l^2}\sum_{l=0}^{L} w_l \sum_{i,j}\left(G_{ij}^l - A_{ij}^l\right)^2. \end{aligned} \tag{6}$$

With random noise as the initial $\mathbf{x}$, Equation (6) can be minimized by gradient descent with backpropagation to generate the final stylized result.

For the implementation issues, [14] uses feature representation provided by VGG network. Other networks which are trained to perform object recognition task are also capable of achieving similar performance (*e.g*., ResNet) and some implementations with optional networks are available online [1]. Moreover, adding a total variation denoising term when implementation will help improve the quality of the generated image.

However, the algorithm of Gatys *et al*. has the limitations of instabilities during iterations. Moreover, it requires manually tuning the parameters, which is very tedious. Risser *et* *al*. argue that the cause of instabilities is that Gram matrices are actually built on feature activations rather than image intensities. However, feature activations with quite different means and variances can still have the same Gram matrix (Figure 4 in [38]). To address the limitation of instabilities, Risser *et al*. introduce histogram losses to add to the objective function in the algorithm of Gatys *et al*. The histogram losses can make the optimization tend to preserve the entire histogram of the feature activations and thus the mean and variance is preserved as well. For the problem of hand-tuned parameters, they propose a simple approach to automatically tune the parameters. With the aid of gradient information, the parameters are automatically tuned so as to prevent extreme values for gradients. The results produced by their algorithm are proved to be more stable and converge better over the iteration count as compare with the original algorithm proposed by Gatys *et al*.

In [30], Li *et al*. further investigate the Neural Style Transfer algorithm of Gatys *et al*. Other than proving that matching the Gram matrices is equivalent to minimizing Maximum Mean Discrepancy (MMD) with the second order polynomial kernel, they also try other kernel functions for MMD in style transfer, including the linear kernel and Gaussian kernel. Their results show that linear kernel achieves comparable results with other kernels and yet requires lower computation complexity. Another finding is that the statistics of Batch Normalization (BN) of a certain layer can also represent the style.

Although aforementioned algorithm achieves remarkable results, they do not consider the semantic content of the image, *i.e*., the transfer process is not content-aware. To tackle this problem, Yin [48] proposes a content-aware Neural Style Transfer algorithm through region segmentation on the basis of the algorithm of Gatys *et al*. Chen and Hsu [10] further investigate content-aware Neural Style Transfer by introducing masking out process to constrain the spatial correspondence and high-order feature statistics so as to improve the result.

2.1.2 MRF-based Descriptive Neural Methods

The Markov Random Field (MRF) is a classic framework for image synthesis. It assumes that the local image patches contain the most relevant statistical dependencies in an image. The second group of Descriptive Neural Methods is based on MRF and considers the Neural Style Transfer at a local level, *i.e*., operating on patches to match the style.

Li and Wand [28] first introduce the idea of MRF into the field of Neural Style Transfer. They find that when calculating the style loss, the algorithm of Gatys *et al*. captures only the per-pixel feature correlations and does not constrain the spatial layout, which leads to the less visual plausibility results for photorealistic styles. Their solution is to introduce a new style loss function E_s which includes a patch-based MRF prior as follows:

$$E_s(\Phi(\mathbf{x}), \Phi(\mathbf{x_s})) = \sum_{i=1}^{m} \left\| \Psi_i(\Phi(\mathbf{x})) - \Psi_{NN(i)}(\Phi(\mathbf{x_s})) \right\|^2, \tag{7}$$

where m is the number of the local patches in stylized image x, Ψ_i denotes the i^{th} local patch and $\Psi_{NN(i)}$ denotes the most similar style patch in the style image $\mathbf{x_s}$ with the i^{th} local patch in the current stylized image $\mathbf{x}$. The best matching $\Psi_{NN(i)}$ is calculated using normalized cross-correlation over all local style patches in $\mathbf{x_s}$.

The algorithm of Li and Wand is actually a content-aware Neural Style Transfer algorithm which, in a sense, considers the semantic content (Figure 7 in [28]). Their design of content-aware transfer makes the results remarkable for photorealistic styles.

Following the work of Li and Wand, Champandard [7] introduces the semantic map into the algorithm. By annotating the input image with a semantic map either manually authored or from the pixel labeling algorithms, the algorithm proposed by Champandard offers the user more control over the stylized result and thus improves the quality of stylization.

2.2. Generative Neural Methods Based On Model Iteration

Although the Descriptive Neural Method is able to yield impressive stylized images, there are still some limitations. One limitation is the efficiency issue. The second category called Generative Neural Methods Based On Model Iteration (also referred to as *"Fast" Neural Style Transfer* in some papers) addresses the speed and computational cost issue at the expense of losing some flexibilities. The key idea is to train a feed-forward network over a large set of images in advance for each specific style image. The network model is optimized by updating the model iteratively using gradient descent.

Johnson *et al*. [24] introduce a fast approach based on the algorithm proposed by Gatys *et al*. They first train an equivalent feed-forward generator network for each specific style. When there is a content image to be stylized, only a single forward pass is required to produce the result. Their presented system consists of two components, *i.e*., image generator network and loss network (Figure 2 in [24]). The architecture of generator network roughly follows the network proposed by Radford *et al*. [37] but with residual blocks as well as strided and fractionally strided convolutions. Rather than using a per-pixel loss function to train the feed-forward network as many image transformation algorithms do, they propose the concept of perceptual loss functions or so-called loss network, which are themselves deep convolutional neural network (VGG-16 network is chosen in [24]). The objective function is similar to the algorithm of Gatys *et al*. but with an extra denoising term.

Another concurrent work of fast generative method is Ulyanov *et al.*'s texture network [44]. The key idea behind the algorithm of Ulyanov *et al*. is similar to the algorithm of Johnson *et al.*, except for the architecture of generator network. Ulyanov *et al*. use a multi-scale architecture as the generator network.

Moveover, inspired by the MRF-based Neural Style Transfer mentioned in Section 2.1, Li and Wand [29] address the efficiency issue by training a Markovian feed-forward network using adversarial training. Their algorithm has a better performance in preserving coherent texture for complex image content, compared with the algorithms proposed by Johnson *et al*. and Ulyanov *et al*.

Although the above generative approaches are capable of producing stylized images two orders of magnitude faster than descriptive methods, their limitations are also obvious. One of the limitations is that each generative network is tied to a single style and separate generative networks have to be trained for each specific style image, which is quite time-consuming and inflexible. A "Faster" Neural Style Transfer algorithm is needed to break these constraints. Dumoulin *et al*. [12] address this issue and propose an algorithm that can learn multiple styles at the same time. Inspired by the fact that some paintings (*e.g*., impressionist paintings) share similar paint strokes but only differ in the color palette, Dumoulin *et al*. believe that many style images may share some computations and it is redundant to train a separate feed-forward network for each of them. Based on the above intuition, they propose an algorithm to train a conditional style transfer network for multiple styles of the same type based on conditional instance normalization. The conditioning is done by scaling and shifting parameters after normalization to each type of style, *i.e*., each style belonging to a specific type of style can be achieved by tuning parameters of an affine transformation after normalization. Furthermore, the algorithm of Dumoulin *et al*. is also capable of combining multiple styles in real-time.

Addressing the same problem as [12], very recently Chen and Schmidt [9] propose a fast patch-based style transfer algorithm. They first propose a fast method based on image iteration in their paper and then, based on the proposed method, they train an inverse network to further speed up this process. For their method based on image iteration, they first extract a set of patches for both content image and style image and swap the activation of each content patch with its closet style patch, which is the process of so-called "swapping the style" in their paper. The activation is constructed in a single layer and thus the computation complexity is reduced and the process is faster. Then based on above algorithm, Chen and Schmidt further train an inverse network to directly invert the swapped activation. Compared with [24, 44, 29], their algorithm is capable of producing stylized image for any new style images with only a single trained inverse network.

3. Slight Modifications of Current Methods

There are several studies presenting some slight modifications based on current state-of-the-art Neural Style Transfer algorithms. These modifications preserve the original architecture and overall process of existing algorithms, but further improve the performance by varying experimental settings, slightly changing the loss term, *etc*.

3.1. Modifications of Descriptive Neural Methods

For the modifications of Descriptive Neural Methods Based On Image Iteration, almost all those modifications are based on the algorithm proposed by Gatys *et al*. in [14]. Novak and Nikulin [34] address the issue that style representation in [14] is invariant to the spatial configuration of the style image and propose a new style representation called "spatial style", which captures less style details and more spatial configurations. Moveover, they also explore the Neural Style Transfer algorithm in [14] by varying different experimental settings (*i.e*., backends, frameworks, networks, initializations points, content layers and style layers), which were not discussed previously. According to their experiments, for the Neural Style Transfer algorithm in [14]: CUDA backend is more stable than OpenCL backend; Torch implementation is faster than Caffe implementation; VGG networks (*i.e*., VGG-16 and VGG-19) have better performance than the AlexNet and GoogLeNet; starting gradient descent from the content image produces better results for most practical applications and starting from while noise helps parameter tuning; relaxing bottom style layer and adding bottom content layer retains the colors of the content image.

In [35], Novak and Nikulin further explore how to improve the Neural Style Transfer algorithm of Gatys *et al*. They find that the algorithm proposed by Gatys *et al*. shows great results in transferring repetitive styles but often fails to transfer complex patterns. Based on this observation, they propose a variety of helpful modifications, including layer weight adjustment, using more layers, activation shift, correlations of features from different layers, correlations and blurred correlations.

Recently Gatys *et al*. [13, 17] themselves propose several slight modifications to improve their previous algorithm. They mainly study three perceptual factors including space, color and scale and demonstrate how to control them during the style transfer. For the spatial control, the strategy is to simply define a guidance map where the desired region (getting the style) is assigned 1 and otherwise, 0. While for the color control, the origin algorithm produces stylized

(a) Content (b) Style (c) Results with Smaller Brush Size (d) Results with Larger Brush Size

Figure 2. Control the brush size in Neural Style Transfer. (c) is the output with smaller brush size and (d) with larger brush size. The style image is "The Starry Night" by Vincent van Gogh.

images with the color distribution of the style image. However, Gatys *et al*. notice that in many cases, people may prefer to preserve the colors of the content image, *i.e*., transfer the style without transferring the colors. They propose two methods to preserve colors in Neural Style Transfer. One method is to first transform the style image's colors to match the content image's colors before style transfer, while the other one is to perform style transfer only in the luminance channel. For the scale control, they separately study the scale control for style mixing and for high resolution (the outputs are with small brush size for high-resolution content image given that the output has the same resolution). The scale control for high resolution is essentially a strategy to control the brush size of the stylized image through a coarse-to-fine procedure with down-sampling and up-sampling. Figure 2 shows the results of our reimplementation of the scale control strategy in controlling the brush size of the outputs. All these strategies make the process of style transfer more controllable and some of these modifications can be generally applied equally to the Generative Neural Methods.

3.2. Modifications of Generative Neural Methods

In practical, the results produced by the Generative Neural Methods are often not as good as the Descriptive Neural Method of Gatys *et al*. To tackle this problem, Ulyanov *et al*. [45, 46] swap batch normalization with instance normalization, which removes instance-specific constrast information from the content image. They find this simple modification can make Generative Neural Methods achieve comparable quality as the Descriptive one. This improvement strategy is also the inspiration of [12] which has been introduced in Section 2.2. Other than instance normalization, in [46], Ulyanov *et al*. also explore how to improve the diversity of the stylized images. They propose a new loss function to encourage the generator to sample unbiasedly from the Julesz texture ensemble.

However, current neural algorithms do not consider the image depth and lose the content image's variation in depth during the process of style transfer. To address this limitation, Liao *et al*. [31] propose a Depth-preserving Neural Style Transfer algorithm. Their approach is to add a depth loss function based on Johnson *et al.*'s Generative Neural Style Transfer algorithm [24]. The depth loss function is to measure the depth difference between the content image and the (yet unknown) stylized image. The depth is acquired by applying the state-of-the-art single-image depth estimation algorithm [52].

4. Extensions to Specific Types of Images

All of aforementioned Neural Style Transfer methods are designed for general still images. They may not be appropriate for other types of images (*e.g*., doodles, head portrait, video frames). Currently, there are many studies aiming to extend state-of-the-art Neural Style Transfer algorithms to these specific types of images as well as single user-specified object stylization.

Neural Style Transfer for Doodles. An interesting extension can be found in [7] by Champandard, which has been introduced in Section 2.1.2. Other than aforementioned contribution which is introducing the semantic map into the Neural Style Transfer algorithm, [7] can also be used to transform a rough sketch into fine artwork using high-level annotations of the input image.

Neural Style Transfer for Head Portrait. Although the algorithm of Gatys *et al*. produces visually plausible results for the stylization of generic images, it is not appropriate for Head Portrait Style Transfer. As it does not impose spatial constraints, directly applying the algorithm of Gatys *et al*. to head portraits often deforms facial structures. Such deformations are unacceptable for human visual system. Selim *et al*. [41] address this problem and extend the algorithm of Gatys *et al*. to head portrait painting transfer. They use the notion of gain maps to constrain spatial configurations, which can transfer the texture of the style image while preserving the facial structures.

Neural Style Transfer for Single User-specified Object. A Targeted Style Transfer algorithm which is the process to stylize only a single user-specified object within an image is proposed by Castillo *et al*. in [5]. The idea is to segment the target object from the stylized image using the state-of-the-art semantic segmentation algorithm and then merge the extracted stylized object with the non-stylized background.

Neural Style Transfer for Video Frames. Another work aimed at extending the Neural Style Transfer algorithm of Gatys *et al*. from still images to video sequences is given in [40] by Ruder *et al.*, which is referred to as Neural Video Style Transfer in our paper. Given a style image, the algorithm of Ruder *et al*. introduces a temporal loss function to transfer its artistic style to the entire video. The key idea behind their algorithm is to use a temporal constraint to preserve smooth transition between individual frames, *i.e*., penalizing deviation along the point trajectories. Their algorithm is shown to be able to eliminate most temporal artifacts and produce smooth stylized videos. Another concurrent work is the algorithm proposed by Anderson *et al*. [3] which is able to render a movie by exploiting optical flow to initialize the style transfer optimization.

5. Evaluation Methodology

There is no ground truth for the problem of Neural Style Transfer. Neural Style Transfer is the creation of art. For the same stylized result, it is common that different people have different or even opposite views. Therefore, the evaluation of visual results produced by Neural Style Transfer algorithms remains an open and important problem.

From our point of view, there are two major types of evaluation methodologies that can be employed in the field of Neural Style Transfer, *i.e*., qualitative evaluation and quantitative evaluation. Qualitative evaluation requires participants to rank the results of different algorithms, and relies on the observation of participants (referred to as "Stylization Perceptual Studies"). The evaluation results may vary depending on multiple factors of participants (*e.g*., age, occupation). Although there are some degrees of uncertainty for qualitative evaluation methodologies, qualitative evaluation can at least provide some information about people's preference of such kind of neural art. While quantitatively evaluation focuses on the precise evaluation index (*e.g*., time complexity) of algorithms.

Currently, in the field of Neural Style Transfer, Generative Neural Methods have become a trending topic in that the speed issue is one of the major concerns for industrial applications. But to the best of our knowledge, none of the previous researches run all of these state-of-the-art Generative Neural Methods under the same experimental settings and compare them both qualitatively and quantitatively. Therefore, in this section, we aim to compare five state-of-the-art Generative Neural Methods, as well as the Descriptive Neural Method of Gatys *et al*. as a reference.

Experimental setup. Totally there are ten style images and twenty content images. All the stylized results are produced by running the codes provided by the authors [43, 23, 27, 19, 8], except for [14]. For [14], we use a popular open source code [22] with some slight modifications according to Section 3. The parameters of all these codes used in our experiment are the default parameters provided by the author of corresponding papers, except for [12, 9]. We use the pre-trained models for [12, 9], which are provided by the author. For all the Generative Neural Methods in our experiment, all the test content images were never observed during the training process.

5.1. Qualitative Evaluation

Stylization perceptual studies. The ultimate test of Neural Style Transfer is how appealing the stylized results are to a human observer. Therefore, for qualitative evaluation of Neural Style Transfer, we propose a stylization perceptual study which is to make human observers to judge and rank the results produced by different algorithms. About 40 observers with different occupations and ages participated in our experiment. Participants in the experiment were shown a series of groups of images. Each group consisted of 8 images of size 512×512 pixels, including one content image, one style image and six stylized images produced by different algorithms (five state-of-the-art Generative Neural Methods and one Descriptive Neural Method). Participants were asked to rank six stylized images according to their own appreciation. For each group, participants were given unlimited time to appreciate and respond.

Evaluation metric of stylization perceptual studies. The average stylization rank score is used as our evaluation metric here. For each group, if one of the stylized results is ranked first by an observer, it will get 6 as a stylization rank score as there are totally six images to rank. Similarly, it gets 5 if it ranks second, 4 if third, *etc*. The final stylization rank score of the stylized result produced by an algorithm is the average score over all observers.

Results. We carefully select six representative groups among our results and show them in Figure 3 as an example. Their corresponding average stylization rank scores are shown in Table 1. Each row represents the results produced by an algorithm and each column represents an image group which has been explained in the beginning of this subsection. From Table 1, we can see that different algorithms have different performances over different styles. In general, the average stylization rank scores of [14, 44, 24] are close, *i.e*., the performances of these three algorithms are relatively similar. This is not unexpected, as [44, 24] exploit similar feature representations with [14]. The ideas behind [44] and [24] are also very similar, except for the architecture of the generator. Some scores of [29, 12, 9] are smaller than 3.00. Since [29] is based on Generative Adversarial Network (GAN), to some extent, there is some uncertainty for the generated results. We believe that for GAN-based Neural Style Transfer method, there is still room for im-

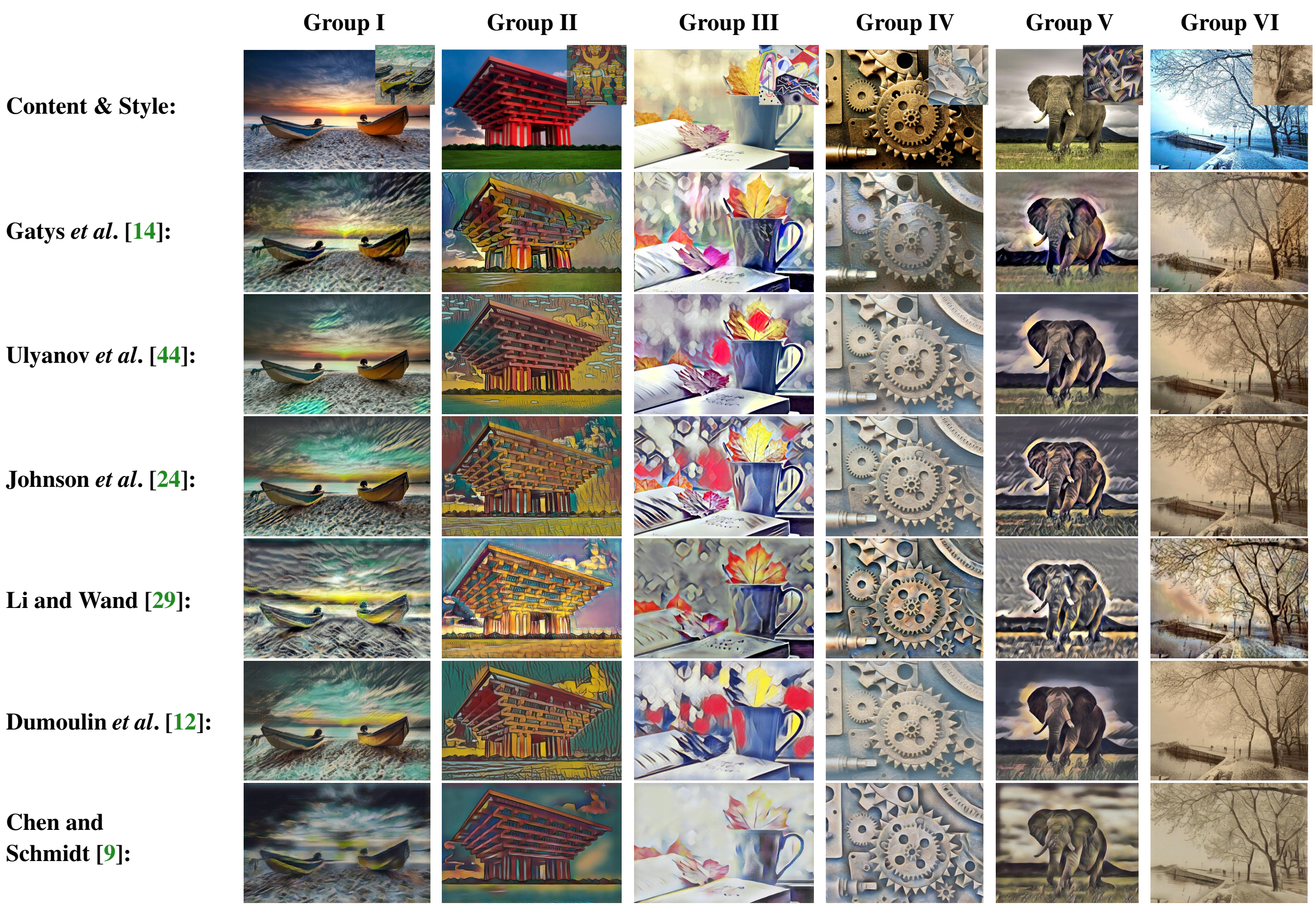

Figure 3. Some example results for qualitative evaluation.

Table 1. Average Stylization Rank Scores ($\in [1, 6]$) of Six Algorithms for Image Groups in Figure 3

Methods	**Average Stylization Rank Score**					
	Group I	Group II	Group III	Group IV	Group V	Group VI
Gatys *et al*. [14]	4.83	3.30	4.20	3.08	3.13	3.15
Ulyanov *et al*. [44]	4.25	4.20	3.85	3.30	4.33	3.63
Johnson *et al*. [24]	4.30	4.15	4.58	4.10	4.15	4.30
Li and Wand [29]	2.70	2.58	3.45	4.25	2.35	2.23
Dumoulin *et al*. [12]	3.40	3.95	2.70	2.43	4.35	3.18
Chen and Schmidt [9]	1.53	2.83	2.23	3.85	2.70	4.53

provement. Although the scores of [12] and [9] are slightly lower than others in our experiment, the results are acceptable in that a three-way trade-off between speed, flexibility and quality is common in research.

5.2. Quantitative Evaluation

Generating time. The efficiency is the focus of Generative Neural Style Transfer algorithms. In this subsection, we compare different algorithms quantitatively in terms of the speed of generating stylized images. Table 2 demonstrates the time required to stylize one image with different resolutions using different algorithms. The last column of Table 2 represents the number of styles that one model of each algorithm can produce. ∞ means that a single model can work for any style. The numbers reported in Table 2 are obtained by averaging the generation time of 100 images.

Table 2. Speed Comparison of Neural Style Transfer Algorithms for One Image of Size 256×256 Pixels, 512×512 Pixels and 1024×1024 Pixels (on a NVIDIA Quadro M6000)

Methods	**Time(s)**			**styles/model**
	256×256	512×512	1024×1024	
Gatys *et al*. [14]	14.32	51.19	200.3	∞
Ulyanov *et al*. [44]	0.026	0.055	0.165	1
Johnson *et al*. [24]	0.012	0.040	0.157	1
Li and Wand [29]	0.017	0.058	0.232	1
Chen and Schmidt [9]	0.142	1.736	2.714	∞

Note: The last column shows the number of styles that a single model can produce.

Note that we do not include the time using the algorithm of Dumoulin *et al*. [12] in Table 2 as their algorithm is to scale and shift parameters based on the algorithm of Johnson *et al*. [24]. Therefore, the time required to stylize one image using [12] is very close to [24] under the same circumstance. For $styles/model$ of [12], the ratio is not fixed and depends on the setting when training the model. From Table 2, we can see that even for high-resolution images, current Generative Neural Methods are still able to stylize them in real-time. Although the generation time for Chen and Schmidt's algorithm [9] is a little longer than the others, their patch-based algorithm is capable of stylizing images for any new styles with a single pre-trained inverse network (*i.e.*, $styles/model = \infty$).

Training time. Another concern for quantitative evaluation is the training time for a single model. The training time of different algorithms is hard to compare as sometimes the model trained for just a few epochs is capable of producing enough visually appealing results. So we just outline the training time of different algorithms during our experiment (under the default setting provided by the author) as a reference for follow-up studies. On a NVIDIA Quadro M6000, the training time for a single model is about 3.5 hours for the algorithm of Johnson *et al*. [24], 3 hours for the algorithm of Ulyanov *et al*. [44] and 2 hours for the algorithm of Li and Wand [29]. The training time for Dumoulin et al.'s algorithm [12] and Chen and Schmidt's algorithm [9] is much longer (*e.g*., a couple of days for [9]), which is acceptable since the value of $styles/model$ of these two algorithms can be much larger than the others.

6. Applications

Due to the amazing stylized results, the research of Neural Style Transfer has led to many successful industrial applications and begun to deliver commercial benefits. There are also some application papers aiming at investigating how to apply Neural Style Transfer technique in different applications [4, 25]. This section summaries these applications and presents some potential usages.

6.1. Social Communication

One of the reasons why Neural Style Transfer catches eyes in both academia and industry is its popularity in some social networking sites (*e.g*., Twitter and Facebook). A mobile application *Prisma* [36] is one of the first industrial applications that provides the Neural Style Transfer algorithm as a service. Before *Prisma*, the general public almost never imagines that one day they are able to turn their photos into art paintings in only a few minutes. Due to its high quality, *Prisma* achieved great success and is becoming popular around the world. Soon some applications providing the same service appeared one after another and began to deliver commercial benefits, *e.g*., a web application *Ostagram* [2] requires users to pay for a faster generating speed. Under the help of these industrial applications, people are able to create their own fantastic art paintings like a painter and share the artwork with others in Twitter and Facebook, which brings a new form of social communication.

The application of Neural Style Transfer in social communication reinforces connections between people and also has positive effects on both academia and industry. For academia, when people share their own masterpiece, they usually make some comments on the disadvantages of the service, which helps the researchers to further improve the algorithm. Moreover, the application of Neural Style Transfer in social communication also drives the advances of other new techniques. For instance, inspired by the real-time requirements of Neural Style Transfer for videos, Facebook AI Research (FAIR) developed a new mobile-embedded deep learning system *Caffe2Go* which can run deep neural networks on mobile phones [21]. For industry, the application brings commercial benefits and promotes the economic development.

6.2. User-assisted Creation Tools

Another use of Neural Style Transfer is to act as user-assisted creation tools. Although, to the best of our knowledge, there are no popular applications that applied the Neural Style Transfer technique in creation tools, we believe that it will be a promising potential usage in the future.

Neural Style Transfer is capable of acting as a creation tool for painters and designers. Neural Style Transfer makes it more convenient for a painter to create an artifact of a specific style, especially when creating computer-made fine art images. Moreover, with Neural Style Transfer algorithms it is trivial to produce stylized fashion elements for fashion designers and stylized CAD drawings for architects in a variety of styles, which is costly to produce them by hand.

6.3. Production Tools for Entertainment Applications

Some entertainment applications such as movies, animations and games are probably the most application forms of Neural Style Transfer. For example, creating an animation usually requires 8 to 24 painted frames per second. The production costs will be largely reduced if Neural Style Transfer can be applied to automatically stylize a live-action video into an animation style. Similarly, Neural Style Transfer can significantly save time and costs when applied to the creation of some movies and computer games.

There are already some application papers aiming at introducing how to apply Neural Style Transfer to the production of movies, *e.g.*, Joshi *et al*. explore the use of Neural Style Transfer in redrawing key scenes in the movie *Come Swim* [25], which indicates the promising potential applications of Neural Style Transfer in this field.

7. Challenges and Possible Solutions

The advances in the field of Neural Style Transfer is amazing and some algorithms have already found use in industrial applications. Although current algorithms achieve remarkable results, there are still several challenges and open issues. In this section, we summarize the challenges within this field of Neural Style Transfer and discuss their corresponding possible solutions.

7.1. Challenges

Problem of parameter tuning. In order to obtain the best results, current Descriptive and Generative Neural Methods usually require a tedious process of manually parameter tuning for each combination of content image and style image. For instance, in the algorithm of Gatys *et al.*, parameters including α, β and w_l in Equation (6) as well as learning rate need carefully tuning to make the results better. For Descriptive Neural Methods, Risser *et al*. introduce their preliminary work on this parameter tuning problem in [38]. Their algorithm simply exploits gradient information to prevent extreme values of gradients. They acknowledge in their paper that the optimization would likely be more mathematically well-founded if the parameter can be tuned with the aid of non-gradient information, *e.g.*, magnitude of the losses or statistics within the losses. We believe that there is still room for improving the automatic parameter tuning strategy for Descriptive Neural Methods. While for Generative Neural Methods, the parameter tuning problem is more serious since a new model needs to be trained for each attempt of parameter tuning. Therefore, the extra time brought by parameter tuning for Generative Neural Methods is much more than Descriptive Neural Methods. However, so far as we know, currently there are no researches focusing on the automatic parameter tuning problem and people usually use the default parameters provided by the author, which are generally capable of producing good results.

Problem of stroke orientation control. Existing Neural Style Transfer algorithms do not consider the brush stroke orientation control in many oil painting styles. However, brush stroke orientation is an important element that can not be ignored in paintings. For instance, brush strokes in Vincent van Gogh's "Starry Night" (Figure 2 (b)) seem to be following a wave which makes the original motionless dark sky seem to move like waves. The brush stroke orientation in paintings is able to impress the viewer and convey the painter's ideas as well as illusions. Nonetheless, the control of stroke orientation is difficult since current Neural Style Transfer algorithms are based on feature statistics. Detailed orientations and continuities of curves in stylized results are hard to guarantee.

Problem of "Fast" and "Faster" Neural Style Transfer. Generative Neural Methods Based On Model Iteration (also referred to as "Fast" and "Faster" Neural Style Transfer) are one of the most important algorithms in the field of Neural Style Transfer, as the high computation costs are often unbearable for industrial applications. Moreover, solving the problem existing in "Fast" and "Faster" Neural Transfer algorithms also benefits the solution of other problems, *e.g.*, parameter tuning problem, which will be introduced in Section 7.2.

Although current "Fast" Neural Style Transfer can generate the results in real-time [24, 44, 29, 45], the flexibility is lost, *i.e.*, a style-specific model needs a considerable amount of time to be trained for each style. The algorithm proposed by Dumoulin *et al*. can learn multiple styles of the same type (*e.g.*, impressionist paintings) at the same time [12]. But their algorithm still needs to train for each style type. While the fast algorithm proposed by Chen and Schmidt is able to work for any new content image and style image [9], the results are not that impressive as others since only a single network layer is exploited. A "Faster" Neural Style Transfer algorithm which can preserve the flexibility and quality simultaneously is greatly in need.

Another problem for "Fast" and "Faster" Neural Style Transfer algorithm is the control of brush stroke size in the output. For Descriptive Neural Methods Based On Image Iteration, the brush stroke size control for high-resolution im-

(a) Content (b) Style (c) Results of Descriptive Method with Brush Size Control (d) Results of Generative Method without Brush Size Control

Figure 4. High-resolution results of Descriptive Neural Method with brush size control and Generative Neural Method without brush size control.

ages can be achieved by using the strategy proposed in [17]. While for Generative Neural Method Based On Model Iteration, the quality of high-resolution results are not quite satisfying (as is shown in Figure 4) and currently no generative methods can adjust the brush stroke size in real-time. Gatys *et al*. declare in [17] that the study of adjusting the brush stroke size for generative methods is trivial as one can simply train on new style images that combines multiple scales. However, the training process is time-consuming and it is redundant to train separate models for different resolutions. Another solution is to resize the content image first and use state-of-the-art super-resolution algorithms [26] to process the stylized result. However, image super-resolution introduces extra computation burden which is not desired. We believe that it is necessary to study a new "Faster" Neural Style Transfer algorithm that can control the brush stroke size in the output in one shot.

7.2. Possible Solutions

Possible solutions to parameter tuning problem. For future research of automatic parameter tuning problem, we discuss the possible solutions for Descriptive Neural Methods and Generative ones separately. For Descriptive Neural Methods, one possible solution is to follow up the work of Risser *et al*. and further combine some non-gradient information such as magnitude of the losses and statistics within the losses. Another direction is to derive some new inspirations from current automatic parameter optimization strategies for classification problems (*e.g.*, Domhan *et al.*'s work in [11] and Luo's work in [32]). While for Generative Neural Methods, one idea is to study a novel method that does not need to train separate models for different styles (just like [9]) and meanwhile, preserves the high quality of the results (*i.e.*, break the three-way trade-off between speed, flexibility and quality). Then the process of parameter tuning will not be that time-consuming and it is acceptable to leave the parameters tuning to users. Moreover, some ideas in current automatic parameter optimization strategies may also be useful for automatic parameter tuning in Generative Neural Methods.

Possible solutions to stroke orientation control problem. Current Neural Style Transfer algorithms do not consider the control of brush stroke orientation. In contrast, in the field of *Non-photorealistic Rendering* (NPR), the control of brush stroke orientation has been well studied (see Chapter 6 in [39] for a review). We believe that some ideas in the NPR field can be borrowed to solve the orientation problem of Neural Style Transfer. For example, Zhang *et al*. require the user to specify where and how the brush stroke orients in [49], since different users have different preferences. The same idea can be borrowed to Neural Style Transfer algorithm, in which the user is asked to select a global stroke orientation in advance. Moreover, combining Neural Style Transfer algorithm and the strategies to guide the orientation in the NPR field (*e.g.*, vector field approach in [50]) is another potential solution for the difficult orientation control problem of Neural Style Transfer.

Possible solutions to problem of "Fast" and "Faster" Neural Style Transfer. The key issue for this research direction is how to break the three-way trade-off between speed, flexibility and quality. One possible solution is to follow up Chen and Schmidt's work [9]. Currently their algorithm is one of the most efficient algorithms but is not that satisfying in terms of image quality. Further improving the quality of the stylized images produced by [9] is a promising direction to break the three-way trade-off between speed, flexibility and quality in the field of Neural Style Transfer. There are already some related work such as [51]. While for the brush stroke size control for "Fast" and "Faster" Neural Style Transfer algorithms, the idea is similar to the previously mentioned possible solution of stroke orientation control. In the field of NPR, there are lots of researches about the brush stroke size control. For a recent review, we recommend Chapter 1 in [39].

8. Conclusions and Future Work

Over the past three years, Neural Style Transfer has continued to become a thriving area of research. Increased activity in this research area has been driven by both scientific challenges and industrial demands. And a considerable

Table 3. Summary of Current Achievements in the Field of Neural Style Transfer

Current Achievements	**Paper**	**Description**
Original Neural Style Transfer as well as its slight modifications	[14, 16]	The First Descriptive Neural Style Transfer algorithm proposed by Gatys *et al*.
	[34, 35]	Slight Modifications by varying experimental settings, *etc*.
Theoretical Explanation of Neural Style Transfer	[30]	Treat Neural Style Transfer as a Domain Adaption problem.
Solution of instabilities during iterations	[38]	Combine [14]'s style loss with a proposed histogram loss.
Automatic parameters tuning	[38]	Exploit gradient information to automatically tune the parameters.
Combining semantic information	[48]	Region Segmentation based Content-aware Neural Style Transfer.
	[10]	Masking Out based Content-aware Neural Style Transfer.
	[28]	Patch-based Markov Random Fields style loss.
	[7]	Combine [28] with the semantic map.
Preserving content image's color	[13, 17]	Color-preserving Neural Style Transfer through color transformation or operating in the luminance channel.
Brush size control in the stylized image	[17]	Coarse-to-fine procedure with down-sampling and up-sampling.
Preserving content image's depth information	[31]	Depth-preserving Neural Style Transfer through introducing an extra depth loss function.
Speeding up time-consuming Descriptive Neural Style Transfer	[24, 44, 29]	"Fast" Generative Neural Style Transfer algorithm through training a style-specific feed-forward network for every style.
	[45, 46]	Improve Generative Neural Methods through swapping Batch Normalization with Instance Normalization and learning generators that uniformly sample the Julesz ensemble.
	[12]	"Faster" Neural Style Transfer through training a conditional network for several styles.
	[9]	"Faster" Neural Style Transfer through training an inverse network for any style.
Turning sketches into artwork	[7]	Neural Doodle algorithm based on Markov Random Fields.
User-specific object stylization	[5]	Targeted Style Transfer through combining Semantic Segmentation algorithm.
Video frames stylization	[40, 3]	Neural Video Style Transfer through enforcing consistency between adjacent frames.
Head portrait stylization	[41]	Head Portrait Style Transfer through exploiting the notion of gain maps.

amount of researches have been conducted in the field of Neural Style Transfer. Key advances in this field are summarized in Table 3. In conclusion, this review provides an extensive survey of existing research efforts on Neural Style Transfer, covering the taxonomy of current methods, their improvements and extensions, evaluation methodology as well as existing challenges and corresponding possible solutions. Moreover, three application domains of Neural Style Transfer are reviewed, including social communication, user-assisted creation tools and production tools for entertainment applications.

Promising directions for future research in Neural Style Transfer mainly focus on two aspects. The first aspect is to solve the existing aforementioned challenges for current algorithms, *i.e*., problem of parameter tuning, problem of stroke orientation control and problem existing in "Fast" and "Faster" Neural Style Transfer algorithms. Descriptions of these challenges as well as corresponding possible solutions have been demonstrated in Section 7. The second aspect of promising directions is to focus on new extensions to Neural Style Transfer (*e.g*., Fashion Style Transfer and Character Style Transfer). There are already some preliminary work related with this direction, such as the recent work of Yang *et al*. [47] on Text Effects Transfer. These interesting extensions may become trending topics in the future and related new areas may be created subsequently.

Acknowledgement

This work is supported by National Natural Science Foundation of China(61572428, U1509206), National Key Research and Development Program (2016YFB1200203), Program of International Science, Technology Cooperation under Grant 2013DFG12840, Fundamental Research Funds for the Central Universities and National High-tech Technology R&D Program (863 Program) of China under Grant 2014AA015205.

References

[1] Neural-style: Using other neural models, 2016. 3

[2] Ostagram, 2016. 1, 9

[3] A. G. Anderson, C. P. Berg, D. P. Mossing, and B. A. Olshausen. DeepMovie: Using optical flow and deep neural networks to stylize movies. *ArXiv e-prints*, May 2016. 7, 12

[4] F. Becattini, A. Ferracani, L. Landucci, D. Pezzatini, T. Uricchio, and A. Del Bimbo. Imaging novecento. a mobile app for automatic recognition of artworks and transfer of artistic styles. In *Euro-Mediterranean Conference*, pages 781–791. Springer, 2016. 9

[5] C. Castillo, S. De, X. Han, B. Singh, A. K. Yadav, and T. Goldstein. Son of zorn's lemma: Targeted style transfer using instance-aware semantic segmentation. *ArXiv e-prints*, Jan. 2017. 6, 12

[6] A. J. Champandard. Deep forger: Paint photos in the style of famous artists, 2015. 1

[7] A. J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *ArXiv e-prints*, Mar. 2016. 4, 6, 12

[8] T. Q. Chen. style-swap. `https://github.com/rtqichen/style-swap`, 2016. 7

[9] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. *ArXiv e-prints*, Dec. 2016. 5, 7, 8, 9, 10, 11, 12

[10] Y.-L. Chen and C.-T. Hsu. Towards deep style transfer: A content-aware perspective. In *Proceedings of the British Machine Vision Conference*, 2016. 4, 12

[11] T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3460–3468, 2015. 11

[12] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. *ArXiv e-prints*, Oct. 2016. 5, 6, 7, 8, 9, 10, 12

[13] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. *ArXiv e-prints*, June 2016. 5, 12

[14] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *ArXiv e-prints*, Aug. 2015. 1, 3, 5, 7, 8, 9, 12

[15] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015. 3

[16] 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*, pages 2414–2423, 2016. 1, 12

[17] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. Controlling perceptual factors in neural style transfer. *ArXiv e-prints*, Nov. 2016. 5, 11, 12

[18] B. Gooch and A. Gooch. *Non-photorealistic rendering*. A. K. Peters, Ltd., Natick, MA, USA, 2001. 1

[19] Google Brain. image_stylization. `https://github.com/tensorflow/magenta/tree/master/magenta/models/image_stylization`, 2016. 7

[20] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012. 2

[21] Y. Jia and P. Vajda. Delivering real-time ai in the palm of your hand, 2016. 9

[22] J. Johnson. neural-style. `https://github.com/jcjohnson/neural-style`, 2015. 7

[23] J. Johnson. fast-neural-style. `https://github.com/jcjohnson/fast-neural-style`, 2016. 7

[24] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711, 2016. 4, 5, 6, 7, 8, 9, 10, 12

[25] B. J. Joshi, K. Stewart, and D. Shapiro. Bringing impressionism to life with neural style transfer in come swim. *ArXiv e-prints*, Jan. 2017. 9, 10

[26] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. 11

[27] C. Li. Mgans. `https://github.com/chuanli11/MGANs`, 2016. 7

[28] 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*, pages 2479–2486, 2016. 4, 12

[29] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716, 2016. 5, 7, 8, 9, 10, 12

[30] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. *ArXiv e-prints*, Jan. 2017. 2, 4, 12

[31] R. Liao, Y. Xia, and X. Zhang. Depth-preserving style transfer. `https://github.com/xiumingzhang/depth-preserving-neural-style-transfer/blob/master/report/egpaper_final.pdf`, 2016. 6, 12

[32] G. Luo. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1):1–16, 2016. 11

[33] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5188–5196, 2015. 3

[34] Y. Nikulin and R. Novak. Exploring the neural algorithm of artistic style. *ArXiv e-prints*, Feb. 2016. 5, 12

[35] R. Novak and Y. Nikulin. Improving the neural algorithm of artistic style. *ArXiv e-prints*, May 2016. 5, 12

[36] I. Prisma Labs. Prisma: Turn memories into art using artificial intelligence, 2016. 1, 9

[37] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ArXiv e-prints*, Nov. 2015. 4

[38] E. Risser, P. Wilmot, and C. Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *ArXiv e-prints*, Jan. 2017. 4, 10, 12

[39] P. Rosin and J. Collomosse. *Image and video-based artistic stylisation*, volume 42. Springer Science & Business Media, 2012. 1, 11

[40] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, pages 26–36, 2016. 7, 12

[41] A. Selim, M. Elgharib, and L. Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics*, 35(4):129, 2016. 6, 12

[42] T. Strothotte and S. Schlechtweg. *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann, 2002. 1

[43] D. Ulyanov. texture_nets. `https://github.com/DmitryUlyanov/texture_nets`, 2016. 7

[44] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *International Conference on Machine Learning*, pages 1349–1357, 2016. 5, 7, 8, 9, 10, 12

[45] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *ArXiv e-prints*, July 2016. 6, 10, 12

[46] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *ArXiv e-prints*, Jan. 2017. 6, 12

[47] S. Yang, J. Liu, Z. Lian, and Z. Guo. Awesome typography: Statistics-based text effects transfer. *ArXiv e-prints*, Nov. 2016. 13

[48] R. Yin. Content-aware neural style transfer. *ArXiv e-prints*, Jan. 2016. 4, 12

[49] E. Zhang, J. Hays, and G. Turk. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(1), 2007. 11

[50] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25(4):1294–1326, 2006. 11

[51] H. Zhang and K. Dana. Multi-style generative network for real-time transfer. *ArXiv e-prints*, Mar. 2017. 11

[52] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 388–396, 2015. 6