Neural Style Transfer: A Review

Yongcheng Jing, Yezhou Yang, *Member, IEEE,* Zunlei Feng, Jingwen Ye,
Yizhou Yu, *Senior Member, IEEE,* and Mingli Song, *Senior Member, IEEE*

Abstract—The seminal work of Gatys et al. demonstrated the power of Convolutional Neural Networks (CNNs) in creating artistic imagery by separating and recombining image content and style. This process of using CNNs to render a content image in different styles is referred to as Neural Style Transfer (NST). Since then, NST has become a trending topic both in academic literature and industrial applications. It is receiving increasing attention and a variety of approaches are proposed to either improve or extend the original NST algorithm. In this paper, we aim to provide a comprehensive overview of the current progress towards NST. We first propose a taxonomy of current algorithms in the field of NST. Then, we present several evaluation methods and compare different NST algorithms both qualitatively and quantitatively. The review concludes with a discussion of various applications of NST and open problems for future research. A list of papers discussed in this review, corresponding codes, pre-trained models and more comparison results are publicly available at: https://github.com/ycjing/Neural-Style-Transfer-Papers.

Index Terms—Neural style transfer (NST), convolutional neural network

1 Introduction

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

Since the mid-1990s, the art theories behind the appealing artworks have been attracting the attention of not only the artists but many computer science researchers. There are plenty of studies and techniques exploring how to automatically turn images into synthetic artworks. Among these studies, the advances in *non-photorealistic rendering* (NPR) [1], [2], [3] are inspiring, and nowadays, it is a firmly established field in the community of computer graphics. However, most of these NPR stylisation algorithms are designed for particular artistic styles [3], [4] and cannot be easily extended to other styles. In the community of computer vision, style transfer is usually studied as a generalised problem of texture synthesis, which is to extract and transfer the texture from the source to target [5], [6], [7], [8]. Hertzmann et al. [9] further propose a framework named *image analogies* to perform a generalised style transfer by learning the analogous transformation from the provided example pairs of unstylised and stylised images. However, the common limitation of these methods is that they only use low-level image features and often fail to capture image structures effectively.

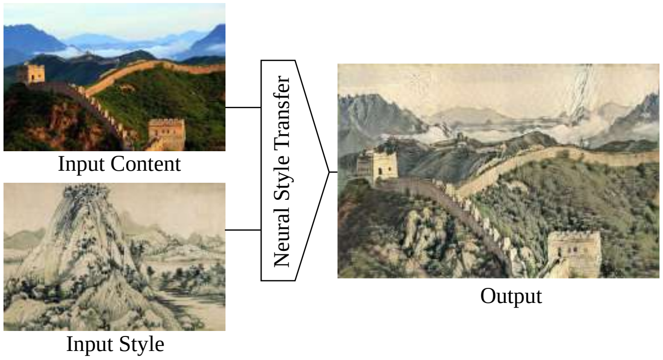


Figure 1: Example of NST algorithm to transfer the style of a Chinese painting onto a given photograph. The style image is named "Dwelling in the Fuchun Mountains" by Gongwang Huang.

Recently, inspired by the power of *Convolutional Neural Networks (CNNs)*, Gatys et al. [10] first studied how to use a CNN to reproduce famous painting styles on natural images. They proposed to model the *content* of a photo as the feature responses from a pre-trained CNN, and further model the *style* of an artwork as the summary feature statistics. Their experimental results demonstrated that a CNN is capable of extracting *content* information from an arbitrary photograph and *style* information from a well-known artwork. Based on this finding, Gatys et al. [10] first proposed to exploit CNN feature activations to recombine the *content* of a given photo and the *style* of famous artworks. The key idea behind their algorithm is to iteratively optimise an image with the objective of matching desired CNN feature distributions, which involves both the photo's *content* information and artwork's *style* information. Their proposed algorithm successfully produces stylised images with the appearance of a given artwork. Figure 1 shows an example of transferring the style of a Chinese painting

- *Y. Jing, Z. Feng, J. Ye, and M. Song are with Microsoft Visual Perception Laboratory, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China. E-mails: {ycjing, zunleifeng, yejingwen, brooksong}@zju.edu.cn.*
- *Y. Yang is with School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA. E-mail: yz.yang@asu.edu.*
- *Y. Yu is with the Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: yizhouy@acm.org.*

"Dwelling in the Fuchun Mountains" onto a photo of The Great Wall. Since the algorithm of Gatys et al. does not have any explicit restrictions on the type of style images and also does not need ground truth results for training, it breaks the constraints of previous approaches. The work of Gatys et al. opened up a new field called *Neural Style Transfer (NST)*, which is the process of using *Convolutional Neural Network* to render a content image in different styles.

The seminal work of Gatys et al. has attracted wide attention from both academia and industry. In academia, lots of follow-up studies were conducted to either improve or extend this NST algorithm. The related researches of NST have also led to many successful industrial applications (e.g., *Prisma* [11], *Ostagram* [12], *Deep Forger* [13]). However, there is no comprehensive survey summarising and discussing recent advances as well as challenges within this new field of Neural Style Transfer.

In this paper, we aim to provide an overview of current advances (up to March 2018) in Neural Style Transfer (NST). Our contributions are threefold. First, we investigate, classify and summarise recent advances in the field of NST. Second, we present several evaluation methods and experimentally compare different NST algorithms. Third, we summarise current challenges in this field and propose possible directions on how to deal with them in future works.

The organisation of this paper is as follows. We start our discussion with a brief review of previous artistic rendering methods without CNNs in Section 2. Then Section 3 explores the derivations and foundations of NST. Based on the discussions in Section 3, we categorise and explain existing NST algorithms in Section 4. Some improvement strategies for these methods and their extensions will be given in Section 5. Section 6 presents several methodologies for evaluating NST algorithms and aims to build a standardised benchmark for follow-up studies. Then we demonstrate the commercial applications of NST in Section 7, including both current successful usages and its potential applications. In Section 8, we summarise current challenges in the field of NST, as well as propose possible directions on how to deal with them in future works. Finally, Section 9 concludes the paper and delineates several promising directions for future research.

2 Style Transfer Without Neural Networks

Artistic stylisation is a long-standing research topic. Due to its wide variety of applications, it has been an important research area for more than two decades. Before the appearance of NST, the related researches have expanded into an area called *non-photorealistic rendering* (NPR). In this section, we briefly review some of these *artistic rendering* (AR) algorithms without CNNs. Specifically, we focus on artistic stylization of 2D images, which is called *image-based artistic rendering* (IB-AR) in [14]. For a more comprehensive overview of IB-AR techniques, we recommend [3], [14], [15]. Following the IB-AR taxonomy defined by Kyprianidis et al. [14], we first introduce each category of IB-AR techniques without CNNs and then discuss their strengths and weaknesses.

Stroke-Based Rendering. Stroke-based rendering (SBR) refers to a process of placing virtual strokes (e.g., brush strokes, tiles, stipples) upon a digital canvas to render a photograph with a particular style [16]. The process of SBR is generally starting from a source photo, incrementally compositing strokes to match the photo, and finally producing a non-photorealistic imagery, which looks like the photo but with an artistic style. During this process, an objective function is designed to guide the greedy or iterative placement of strokes.

The goal of SBR algorithms is to faithfully depict a prescribed style. Therefore, they are generally effective at simulating certain types of styles (e.g., oil paintings, watercolours, sketches). However, each SBR algorithm is carefully designed for only one particular style and not capable of simulating an arbitrary style, which is not flexible.

Region-Based Techniques. Region-based rendering is to incorporate region segmentation to enable the adaption of rendering based on the content in regions. Early region-based IB-AR algorithms exploit the shape of regions to guide the stroke placement [17], [18]. In this way, different stroke patterns can be produced in different semantic regions in an image. Song et al. [19] further propose a region-based IB-AR algorithm to manipulate geometry for artistic styles. Their algorithm creates simplified shape rendering effects by replacing regions with several canonical shapes.

Considering regions in rendering allows the local control over the level of details. However, the problem in SBR persists: one region-based rendering algorithm is not capable of simulating an arbitrary style.

Example-Based Rendering. The goal of example-based rendering is to learn the mapping between an exemplar pair. This category of IB-AR techniques is pioneered by Hertzmann et al., who propose a framework named image analogies [9]. Image analogies aim to learn a mapping between a pair of source images and target stylised images in a supervised manner. The training set of image analogy comprises pairs of unstylised source images and the corresponding stylised images with a particular style. Image analogy algorithm then learns the analogous transformation from the example training pairs and creates analogous stylised results when given a test input photograph. Image analogy can also be extended in various ways, e.g., to learn stroke placements for portrait painting rendering [20].

In general, image analogies are effective for a variety of artistic styles. However, pairs of training data are usually unavailable in practice. Another limitation is that image analogies only exploit low-level image features. Therefore, image analogies typically fail to effectively capture content and style, which limits the performance.

Image Processing and Filtering. Creating an artistic image is a process that aims for image simplification and abstraction. Therefore, it is natural to consider adopting and combining some related image processing filters to render a given photo. For example, in [21], Winnemöller et al. for the first time exploit bilateral [22] and difference of Gaussians filters [23] to automatically produce cartoon-like effects.

Compared with other categories of IB-AR techniques, image-filtering based rendering algorithms are generally straightforward to implement and efficient in practice. At an expense, they are very limited in style diversity.

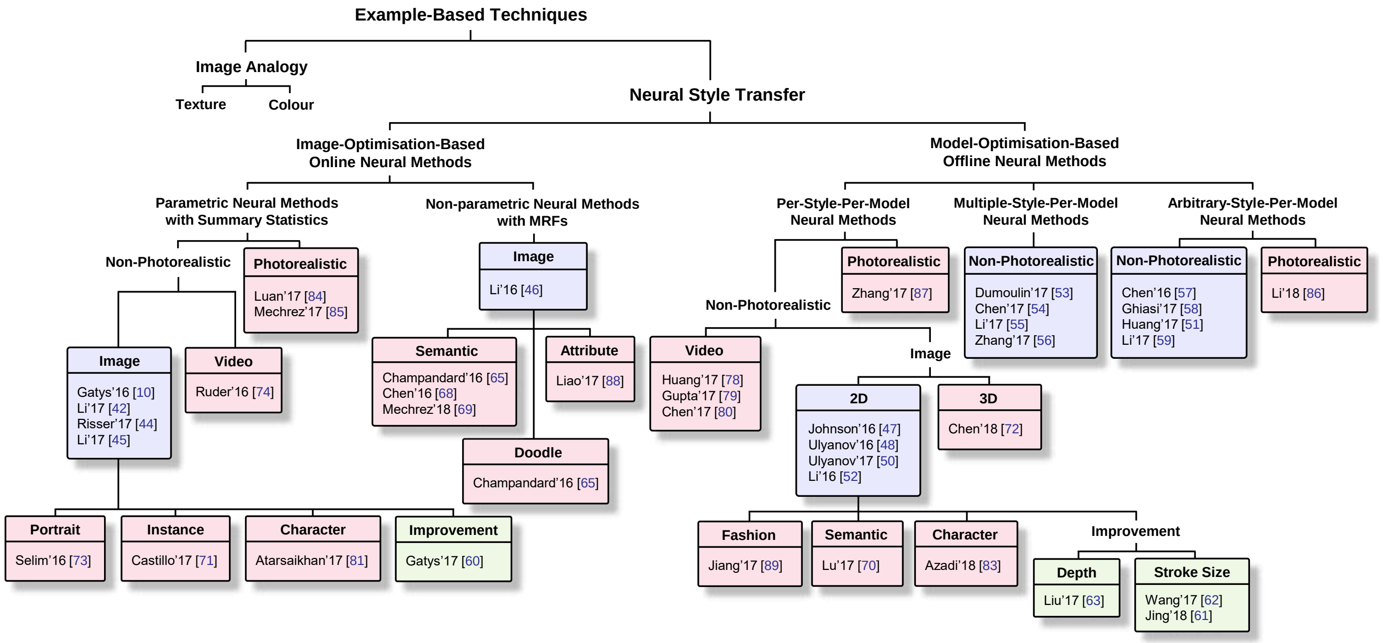


Figure 2: A taxonomy of NST techniques. Our proposed NST taxonomy extends the IB-AR taxonomy proposed by Kyprianidis et al. [14].

Summary. Based on the above discussions, although some IB-AR algorithms without CNNs are capable of faithfully depicting certain prescribed styles, they typically have the limitations in flexibility, style diversity, and effective image structure extractions. Therefore, there is a demand for novel algorithms to address these limitations, which gives birth to the field of NST.

3 Derivations of Neural Style Transfer

For a better understanding of the NST development, we start by introducing its derivations. To automatically transfer an artistic style, the first and most important issue is how to model and extract *style* from an image. Since *style* is very related to *texture*[1], a straightforward way is to relate *Visual Style Modelling* back to previously well-studied *Visual Texture Modelling* methods. After obtaining the style representation, the next issue is how to reconstruct an image with desired style information while preserving its content, which is addressed by the *Image Reconstruction* techniques.

3.1 Visual Texture Modelling

Visual texture modelling [24] is previously studied as the heart of texture synthesis [25], [26]. Throughout the history, there are two distinct approaches to model visual textures, which are *Parametric Texture Modelling with Summary Statistics* and *Non-parametric Texture Modelling with Markov Random Fields (MRFs)*.

1) Parametric Texture Modelling with Summary Statistics. One path towards texture modelling is to capture image statistics from a sample texture and exploit summary statistical property to model the texture. The idea is first proposed by Julesz [27], who models textures as pixel-based N-th order statistics. Later, the work in [28] exploits filter responses to analyze textures, instead of direct pixel-based measurements. After that, Portilla and Simoncelli [29] further introduce a texture model based on multi-scale orientated filter responses and use gradient descent to improve synthesised results. A more recent parametric texture modelling approach proposed by Gatys et al. [30] is the first to measure summary statistics in the domain of a CNN. They design a Gram-based representation to model textures, which is the correlations between filter responses in different layers of a pre-trained classification network (VGG network) [31]. More specifically, the Gram-based representation encodes the second order statistics of the set of CNN filter responses. Next, we will explain this representation in detail for the usage of the following sections.

Assume that the feature map of a sample texture image I_s at layer l of a pre-trained deep classification network is $\mathcal{F}^l(I_s) \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels, and H and W represent the height and width of the feature map $\mathcal{F}(I_s)$. Then the Gram-based representation can be obtained by computing the Gram matrix $\mathcal{G}(\mathcal{F}^l(I_s)') \in \mathbb{R}^{C\times C}$ over the feature map $\mathcal{F}^l(I_s)' \in \mathbb{R}^{C\times (HW)}$ (a reshaped version of $\mathcal{F}^l(I_s)$):

$$\mathcal{G}(\mathcal{F}^l(I_s)') = [\mathcal{F}^l(I_s)'][\mathcal{F}^l(I_s)']^T. \quad (1)$$

This Gram-based texture representation from a CNN is effective at modelling wide varieties of both natural and non-natural textures. However, the Gram-based representation is designed to capture global statistics and tosses spatial arrangements, which leads to unsatisfying results for modelling regular textures with long-range symmetric structures. To address this problem, Berger and Memisevic [32] propose to horizontally and vertically translate feature

[1] We clarify that style is very related to texture but not limited to texture. Style also involves a large degree of simplification and shape abstraction effects, which falls back to the composition or alignment of texture features.

maps by δ pixels to correlate the feature at position (i, j) with those at positions $(i + \delta, j)$ and $(i, j + \delta)$. In this way, the representation incorporates spatial arrangement information and is therefore more effective at modelling textures with symmetric properties.

2) Non-parametric Texture Modelling with MRFs. Another notable texture modelling methodology is to use non-parametric resampling. A variety of non-parametric methods are based on MRFs model, which assumes that in a texture image, each pixel is entirely characterised by its spatial neighbourhood. Under this assumption, Efros and Leung [25] propose to synthesise each pixel one by one by searching similar neighbourhoods in the source texture image and assigning the corresponding pixel. Their work is one of the earliest non-parametric algorithms with MRFs. Following their work, Wei and Levoy [26] further speed up the neighbourhood matching process by always using a fixed neighbourhood.

3.2 Image Reconstruction

In general, an essential step for many vision tasks is to extract an abstract representation from the input image. Image reconstruction is a reverse process, which is to reconstruct the whole input image from the extracted image representation. It is previously studied to analyse a particular image representation and discover what information is contained in the abstract representation. Here our major focus is on CNN representation based image reconstruction algorithms, which can be categorised into *Image-Optimisation-Based Online Image Reconstruction* (IOB-IR) and *Model-Optimisation-Based Offline Image Reconstruction* (MOB-IR).

1) Image-Optimisation-Based Online Image Reconstruction. The first algorithm to reverse CNN representations is proposed by Mahendran and Vedaldi [33], [34]. Given a CNN representation to be reversed, their algorithm iteratively optimises an image (generally starting from random noise) until it has a similar desired CNN representation. The iterative optimisation process is based on gradient descent in image space. Therefore, the process is time-consuming especially when the desired reconstructed image is large.

2) Model-Optimisation-Based Offline Image Reconstruction. To address the efficiency issue of [33], [34], Dosovitskiy and Brox [35] propose to train a feed-forward network in advance and put the computational burden at training stage. At testing stage, the reverse process can be simply done with a network forward pass. Their algorithm significantly speeds up the image reconstruction process. In their later work [36], they further combine Generative Adversarial Network (GAN) [37] to improve the results.

4 A Taxonomy of Neural Style Transfer Algorithms

NST is a subset of the aforementioned example-based IB-AR techniques. In this section, we first provide a categorisation of NST algorithms and then explain major 2D image based non-photorealistic NST algorithms (Figure 2, purple boxes) in detail. More specifically, for each algorithm, we start by introducing the main idea and then discuss its weaknesses and strengths. Since it is complex to define the notion of style [3], [38] and therefore very subjective to define what criteria are important to make a successful style transfer algorithm [39], here we try to evaluate these algorithms in a more structural way by only focusing on *details, semantics, depth and variations in brush strokes*[2]. We will discuss more about the problem of aesthetic evaluation criterion in Section 8 and also present more evaluation results in Section 6.

Our proposed taxonomy of NST techniques is shown in Figure 2. We keep the taxonomy of IB-AR techniques proposed by Kyprianidis et al. [14] unaffected and extend it by NST algorithms. Current NST methods fit into one of two categories, *Image-Optimisation-Based Online Neural Methods* (IOB-NST) and *Model-Optimisation-Based Offline Neural Methods* (MOB-NST). The first category transfers the style by iteratively optimising an image, i.e., algorithms belong to this category are built upon IOB-IR techniques. The second category optimises a generative model offline and produces the stylised image with a single forward pass, which exploits the idea of MOB-IR techniques.

4.1 Image-Optimisation-Based Online Neural Methods

DeepDream [40] is the first attempt to produce artistic images by reversing CNN representations with IOB-IR techniques. By further combining *Visual Texture Modelling* techniques to model style, IOB-NST algorithms are subsequently proposed, which build the early foundations for the field of NST. Their basic idea is to first model and extract style and content information from the corresponding style and content images, recombine them as the target representation, and then iteratively reconstruct a stylised result that matches the target representation. In general, different IOB-NST algorithms share the same IOB-IR technique, but differ in the way they model the visual style, which is built on the aforementioned two categories of *Visual Texture Modelling* techniques. The common limitation of IOB-NST algorithms is that they are computationally expensive, due to the iterative image optimisation procedure.

4.1.1 Parametric Neural Methods with Summary Statistics

The first subset of IOB-NST methods is based on *Parametric Texture Modelling with Summary Statistics*. The style is characterised as a set of spatial summary statistics.

We start by introducing the first NST algorithm proposed by Gatys et al. [4], [10]. By reconstructing representations from intermediate layers of the VGG-19 network, Gatys et al. observe that a deep convolutional neural network is capable of extracting 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 the newly stylised image by penalising the difference of high-level representations derived from content and stylised images, and further build the style component by matching Gram-based summary statistics of style and stylised images, which is derived from their proposed texture modelling technique [30] (Section 3.1). The details of their algorithm are as follows.

[2] We claim that the visual criteria with respect to a successful style transfer are definitely not limited to these factors.

Given a content image I_c and a style image I_s, the algorithm in [4] tries to seek a stylised image I that minimises the following objective:

$$\begin{aligned} I^* &= \arg\min_I \mathcal{L}_{total}(I_c, I_s, I) \\ &= \arg\min_I \ \alpha\mathcal{L}_c(I_c, I) + \beta\mathcal{L}_s(I_s, I), \end{aligned} \tag{2}$$

where $\mathcal{L}_c$ compares the content representation of a given content image to that of the stylised image, and $\mathcal{L}_s$ compares the Gram-based style representation derived from a style image to that of the stylised image. α and β are used to balance the content component and style component in the stylised result.

The content loss $\mathcal{L}_c$ is defined by the squared Euclidean distance between the feature representations $\mathcal{F}^l$ of the content image I_c in layer l and that of the stylised image I which is initialised with a noise image:

$$\mathcal{L}_c = \sum\nolimits_{l\in\{l_c\}} \|\mathcal{F}^l(I_c) - \mathcal{F}^l(I)\|^2, \tag{3}$$

where $\{l_c\}$ denotes the set of VGG layers for computing the content loss. For the style loss $\mathcal{L}_s$, [4] exploits Gram-based visual texture modelling technique to model the style, which has already been explained in Section 3.1. Therefore, the style loss is defined by the squared Euclidean distance between the Gram-based style representations of I_s and I:

$$\mathcal{L}_s = \sum\nolimits_{l\in\{l_s\}} \|\mathcal{G}(\mathcal{F}^l(I_s)') - \mathcal{G}(\mathcal{F}^l(I)')\|^2, \tag{4}$$

where $\mathcal{G}$ is the aforementioned Gram matrix to encode the second order statistics of the set of filter responses. $\{l_s\}$ represents the set of VGG layers for calculating the style loss.

The choice of content and style layers is an important factor in the process of style transfer. Different positions and numbers of layers can result in very different visual experiences. Given the pre-trained VGG-19 [31] as the loss network, Gatys et al.'s choice of $\{l_s\}$ and $\{l_c\}$ in [4] is $\{l_s\} = \{relu1\_1, relu2\_1, relu3\_1, relu4\_1, relu5\_1\}$ and $\{l_c\} = \{relu4\_2\}$. For $\{l_s\}$, the idea of combining multiple layers (up to higher layers) is critical for the success of Gatys et al.'s NST algorithm. Matching the multi-scale style representations leads to a smoother and more continuous stylisation, which gives the visually most appealing results [4]. For the content layer $\{l_c\}$, matching the content representations on a lower layer preserves the undesired fine structures (e.g., edges and colour map) of the original content image during stylisation. In contrast, by matching the content on a higher layer of the network, the fine structures can be altered to agree with the desired style while preserving the content information of the content image. Also, using VGG-based loss networks for style transfer is not the only option. Similar performance can be achieved by selecting other pre-trained classification networks, e.g., ResNet [41].

In Equation (2), both $\mathcal{L}_c$ and $\mathcal{L}_s$ are differentiable. Thus, with random noise as the initial I, Equation (2) can be minimised by using gradient descent in image space with backpropagation. In addition, a total variation denoising term is usually added in practice to encourage the smoothness in the stylised result.

The algorithm of Gatys et al. does not need ground truth data for training and also does not have explicit restrictions on the type of style images, which addresses the limitations of previous IB-AR algorithms without CNNs (Section 2). However, the algorithm of Gatys et al. does not perform well in preserving the coherence of fine structures and details during stylisation since CNN features inevitably lose some low-level information. Also, it generally fails for photorealistic synthesis, due to the limitations of Gram-based style representation. Moreover, it does not consider the variations of brush strokes and the semantics and depth information contained in the content image, which are important factors in evaluating the visual quality.

In addition, a Gram-based style representation is not the only choice to statistically encode style information. There are also some other effective statistical style representations, which are derived from a Gram-based representation. Li et al. [42] derive some different style representations by considering style transfer in the domain of transfer learning, or more specifically, *domain adaption* [43]. Given that training and testing data are drawn from different distributions, the goal of domain adaption is to adapt a model trained on labelled training data from a source domain to predict labels of unlabelled testing data from a target domain. One way for domain adaption is to match a sample in the source domain to that in the target domain by minimising their distribution discrepancy, in which *Maximum Mean Discrepancy (MMD)* is a popular choice to measure the discrepancy between two distributions. Li et al. prove that matching Gram-based style representations between a pair of style and stylised images is intrinsically minimising MMD with a quadratic polynomial kernel. Therefore, it is expected that other kernel functions for MMD can be equally applied in NST, e.g., the linear kernel, polynomial kernel and Gaussian kernel. Another related representation is the batch normalisation (BN) statistic representation, which is to use mean and variance of the feature maps in VGG layers to model style:

$$\begin{aligned} \mathcal{L}_s = \sum\nolimits_{l\in\{l_s\}} \frac{1}{C^l} \sum_{c=1}^{C^l} \|\mu(\mathcal{F}^l_c(I_s)) - \mu(\mathcal{F}^l_c(I))\|^2 + \\ \|\sigma(\mathcal{F}^l_c(I_s)) - \sigma(\mathcal{F}^l_c(I))\|^2, \end{aligned} \tag{5}$$

where $\mathcal{F}^l_c \in \mathbb{R}^{H\times W}$ is the c-th feature map channel at layer l of VGG network, and C^l is the number of channels.

The main contribution of Li et al.'s algorithm is to theoretically demonstrate that the Gram matrices matching process in NST is equivalent to minimising MMD with the second order polynomial kernel, thus proposing a timely interpretation of NST and making the principle of NST clearer. However, the algorithm of Li et al. does not resolve the aforementioned limitations of Gatys et al.'s algorithm.

One limitation of the Gram-based algorithm is its instabilities during optimisations. Also, it requires manually tuning the parameters, which is very tedious. Risser et al. [44] find that feature activations with quite different means and variances can still have the same Gram matrix, which is the main reason of instabilities. Inspired by this observation, Risser et al. introduce an extra histogram loss, which guides the optimisation to match the entire histogram of feature activations. They also present a preliminary solution to automatic parameter tuning, which is to explicitly prevent gradients with extreme values through extreme gradient normalisation.

By additionally matching the histogram of feature activations, the algorithm of Risser et al. achieves a more stable style transfer with fewer iterations and parameter tuning efforts. However, its benefit comes at an expense of a high computational complexity. Also, the aforementioned weaknesses of Gatys et al.'s algorithm still exist, e.g., a lack of consideration in depth and the coherence of details.

All these aforementioned neural methods only compare content and stylised images in the CNN feature space to make the stylised image semantically similar to the content image. But since CNN features inevitably lose some low-level information contained in the image, there are usually some unappealing distorted structures and irregular artefacts in the stylised results. To preserve the coherence of fine structures during stylisation, Li et al. [45] propose to incorporate additional constraints upon low-level features in pixel space. They introduce an additional Laplacian loss, which is defined as the squared Euclidean distance between the Laplacian filter responses of a content image and stylised result. Laplacian filter computes the second order derivatives of the pixels in an image and is widely used for edge detection.

The algorithm of Li et al. has a good performance in preserving the fine structures and details during stylisation. But it still lacks considerations in semantics, depth, variations in brush strokes, etc.

4.1.2 Non-parametric Neural Methods with MRFs

Non-parametric IOB-NST is built on the basis of *Non-parametric Texture Modelling with MRFs*. This category considers NST at a local level, i.e., operating on patches to match the style.

Li and Wand [46] are the first to propose an MRF-based NST algorithm. They find that the parametric NST method with summary statistics only captures the per-pixel feature correlations and does not constrain the spatial layout, which leads to a less visually plausible result for photorealistic styles. Their solution is to model the style in a non-parametric way and introduce a new style loss function which includes a patch-based MRF prior:

$$\mathcal{L}_s = \sum\nolimits_{l \in \{l_s\}} \sum_{i=1}^{m} \|\Psi_i(\mathcal{F}^l(I)) - \Psi_{NN(i)}(\mathcal{F}^l(I_s))\|^2, \quad (6)$$

where $\Psi(\mathcal{F}^l(I))$ is the set of all local patches from the feature map $\mathcal{F}^l(I)$. Ψ_i denotes the i^{th} local patch and $\Psi_{NN(i)}$ is the most similar style patch with the i-th local patch in the stylised image I. The best matching $\Psi_{NN(i)}$ is obtained by calculating normalised cross-correlation over all style patches in the style image I_s. m is the total number of local patches. Since their algorithm matches a style in the patch-level, the fine structure and arrangement can be preserved much better.

The advantage of the algorithm of Li and Wand is that it performs especially well for photorealistic styles, or more specifically, when the content photo and the style are similar in shape and perspective, due to the patch-based MRF loss. However, it generally fails when the content and style images have strong differences in perspective and structure since the image patches could not be correctly matched. It is also limited in preserving sharp details and depth information.

4.2 Model-Optimisation-Based Offline Neural Methods

Although IOB-NST is able to yield impressive stylised images, there are still some limitations. The most concerned limitation is the efficiency issue. The second category MOB-NST addresses the speed and computational cost issue by exploiting MOB-IR to reconstruct the stylised result, i.e., a feed-forward network g is optimised over a large set of images I_c for one or more style images I_s:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{total}(I_c, I_s, g_{\theta^*}(I_c)), \; I^* = g_{\theta^*}(I_c). \quad (7)$$

Depending on the number of artistic styles a single g can produce, MOB-NST algorithms are further divided into *Per-Style-Per-Model* (PSPM) MOB-NST methods , *Multiple-Style-Per-Model* (MSPM) MOB-NST Methods, and *Arbitrary-Style-Per-Model* (ASPM) MOB-NST Methods.

4.2.1 Per-Style-Per-Model Neural Methods

1) Parametric PSPM with Summary Statistics. The first two MOB-NST algorithms are proposed by Johnson et al. [47] and Ulyanov et al. [48] respectively. These two methods share a similar idea, which is to pre-train a feed-forward style-specific network and produce a stylised result with a single forward pass at testing stage. They only differ in the network architecture, for which Johnson et al. 's design roughly follows the network proposed by Radford et al. [49] but with residual blocks as well as fractionally strided convolutions, and Ulyanov et al. use a multi-scale architecture as the generator network. The objective function is similar to the algorithm of Gatys et al. [4], which indicates that they are also *Parametric Methods with Summary Statistics*.

The algorithms of Johnson et al. and Ulyanov et al. achieve a real-time style transfer. However, their algorithm design basically follows the algorithm of Gatys et al., which makes them suffer from the same aforementioned issues as Gatys et al.'s algorithm (e.g., a lack of consideration in the coherence of details and depth information).

Shortly after [47], [48], Ulyanov et al. [50] further find that simply applying normalisation to every single image rather than a batch of images (precisely *batch normalization (BN)*) leads to a significant improvement in stylisation quality. This single image normalisation is called *instance normalisation* (IN), which is equivalent to batch normalisation when the batch size is set to 1. The style transfer network with IN is shown to converge faster than BN and also achieves visually better results. One interpretation is that IN is a form of style normalisation and can directly normalise the style of each content image to the desired style [51]. Therefore, the objective is easier to learn as the rest of the network only needs to take care of the content loss.

2) Non-parametric PSPM with MRFs. Another work by Li and Wand [52] is inspired by the MRF-based NST [46] algorithm in Section 4.1.2. They address the efficiency issue by training a Markovian feed-forward network using adversarial training. Similar to [46], their algorithm is a *Patch-based Non-parametric Method with MRFs*. Their method is shown to outperform the algorithms of Johnson et al. and Ulyanov et al. in the preservation of coherent textures in complex images, thanks to their patch-based design. However, their algorithm has a less satisfying performance with non-texture styles (e.g., face images), since their algorithm lacks a consideration in semantics. Other weaknesses of their algorithm include a lack of consideration in depth information and variations of brush strokes, which are important visual factors.

4.2.2 *Multiple-Style-Per-Model Neural Methods*

Although the above PSPM approaches can produce stylised images two orders of magnitude faster than previous IOB-NST methods, separate generative networks have to be trained for each particular style image, which is quite time-consuming and inflexible. But many paintings (e.g., impressionist paintings) share similar paint strokes and only differ in their colour palettes. Intuitively, it is redundant to train a separate network for each of them. MSPM is therefore proposed, which improves the flexibility of PSPM by further incorporating multiple styles into one single model. There are generally two paths towards handling this problem: 1) tying only a small number of parameters in a network to each style ([53], [54]) and 2) still exploiting only a single network like PSPM but combining both style and content as inputs ([55], [56]).

1) Tying only a small number of parameters to each style. An early work by Dumoulin et al. [53] is built on the basis of the proposed IN layer in PSPM algorithm [50] (Section 4.2.1). They surprisingly find that using the same convolutional parameters but only scaling and shifting parameters in IN layers is sufficient to model different styles. Therefore, they propose an algorithm to train a conditional multi-style transfer network based on conditional instance normalisation (CIN), which is defined as:

$$\mathrm{CIN}(\mathcal{F}(I_c), s) = \gamma^s \left(\frac{\mathcal{F}(I_c) - \mu(\mathcal{F}(I_c))}{\sigma(\mathcal{F}(I_c))} \right) + \beta^s, \qquad (8)$$

where $\mathcal{F}$ is the input feature activation and s is the index of the desired style from a set of style images. As shown in Equation (8), the conditioning for each style I_s is done by scaling and shifting parameters γ^s and β^s after normalising feature activation $\mathcal{F}(I_c)$, i.e., each style I_s can be achieved by tuning parameters of an affine transformation. The interpretation is similar to that for [50] in Section 4.2.1, i.e., the normalisation of feature statistics with different affine parameters can normalise input content image to different styles. Furthermore, the algorithm of Dumoulin et al. can also be extended to combine multiple styles in a single stylised result by combining affine parameters of different styles.

Another algorithm which follows the first path of MSPM is proposed by Chen et al. [54]. Their idea is to explicitly decouple style and content, i.e., using separate network components to learn the corresponding content and style information. More specifically, they use mid-level convolutional filters (called "StyleBank" layer) to individually learn different styles. Each style is tied to a set of parameters in "StyleBank" layer. The rest components in the network are used to learn content information, which is shared by different styles. Their algorithm also supports flexible incremental training, which is to fix the content components in the network and only train a "StyleBank" layer for a new style.

In summary, both the algorithms of Dumoulin et al. and Chen et al. have the benefits of little efforts needed to learn a new style and a flexible control over style fusion. However, they do not address the common limitations of NST algorithms, e.g., a lack of details, semantics, depth and variations in brush strokes.

2) Combining both style and content as inputs. One disadvantage of the first category is that the model size generally becomes larger with the increase of the number of learned styles. The second path of MSPM addresses this limitation by fully exploring the capability of one single network and combining both content and style into the network for style identification. Different MSPM algorithms differ in the way to incorporate style into the network.

In [55], given N target styles, Li et al. design a selection unit for style selection, which is a N-dimensional one-hot vector. Each bit in the selection unit represents a specific style I_s in the set of target styles. For each bit in the selection unit, Li et al. first sample a corresponding noise map $f(I_s)$ from a uniform distribution and then feed $f(I_s)$ into the style sub-network to obtain the corresponding style encoded features $\mathcal{F}(f(I_s))$. By feeding the concatenation of the style encoded features $\mathcal{F}(f(I_s))$ and the content encoded features $Enc(I_c)$ into the decoder part Dec of the style transfer network, the desired stylised result can be produced: $I = Dec(\ \mathcal{F}(f(I_s)) \ \oplus \ Enc(I_c)\ )$.

Another work by Zhang and Dana [56] first forwards each style image in the style set through the pre-trained VGG network and obtain multi-scale feature activations $\mathcal{F}(I_s)$ in different VGG layers. Then multi-scale $\mathcal{F}(I_s)$ are combined with multi-scale encoded features $Enc(I_c)$ from different layers in the encoder through their proposed inspiration layers. The inspiration layers are designed to reshape $\mathcal{F}(I_s)$ to match the desired dimension, and also have a learnable weight matrix to tune feature maps to help minimise the objective function.

The second type of MSPM addresses the limitation of the increased model size in the first type of MSPM. At an expense, the style scalability of the second type of MSPM is much smaller, since only one single network is used for multiple styles. We will quantitatively compare the style scalability of different MSPM algorithms in Section 6. In addition, some aforementioned limitations in the first type of MSPM still exist, i.e., the second type of MSPM algorithms are still limited in preserving the coherence of fine structures and also depth information.

4.2.3 *Arbitrary-Style-Per-Model Neural Methods*

The third category, ASPM-MOB-NST, aims at one-model-for-all, i.e., one single trainable model to transfer arbitrary artistic styles. There are also two types of ASPM, one built upon *Non-parametric Texture Modelling with MRFs* and the other one built upon *Parametric Texture Modelling with Summary Statistics*.

1) Non-parametric ASPM with MRFs. The first ASPM algorithm is proposed by Chen and Schmidt [57]. They first extract a set of activation patches from content and style feature activations computed in pre-trained VGG network. Then they match each content patch to the most similar style patch and swap them (called "Style Swap" in [57]). The stylised result can be produced by reconstructing the resulting activation map after "Style Swap", with either IOB-IR or MOB-IR techniques. The algorithm of Chen and

Schmidt is more flexible than the previous approaches due to its characteristic of one-model-for-all-style. But the stylised results of [57] are less appealing since the content patches are typically swapped with the style patches which are not representative of the desired style. As a result, the content is well preserved while the style is generally not well reflected.

2) Parametric ASPM with Summary Statistics. Considering [53] in Section 4.2.2, the simplest approach for arbitrary style transfer is to train a separate parameter prediction network P to predict γ^s and β^s in Equation (8) with a number of training styles [58]. Given a test style image I_s, CIN layers in the style transfer network take affine parameters γ^s and β^s from $P(I_s)$, and normalise the input content image to the desired style with a forward pass.

Another similar approach based on [53] is proposed by Huang and Belongie [51]. Instead of training a parameter prediction network, Huang and Belongie propose to modify conditional instance normalisation (CIN) in Equation (8) to adaptive instance normalisation (AdaIN):

$$\begin{aligned}\text{AdaIN}(\mathcal{F}(I_c), \mathcal{F}(I_s)) = \\ \sigma(\mathcal{F}(I_s)) \left(\frac{\mathcal{F}(I_c) - \mu(\mathcal{F}(I_c))}{\sigma(\mathcal{F}(I_c))} \right) + \mu(\mathcal{F}(I_s)).\end{aligned} \tag{9}$$

AdaIN transfers the channel-wise mean and variance feature statistics between content and style feature activations, which also shares a similar idea with [57]. Different from [53], the encoder in the style transfer network of [51] is fixed and comprises the first few layers in pre-trained VGG network. Therefore, $\mathcal{F}$ in [51] is the feature activation from a pre-trained VGG network. The decoder part needs to be trained with a large set of style and content images to decode resulting feature activations after AdaIN to the stylised result: $I = Dec(\ \text{AdaIN}(\mathcal{F}(I_c), \mathcal{F}(I_s))\ )$.

The algorithm of Huang and Belongie [51] is the first ASPM algorithm that achieves a real-time stylisation. However, the algorithm of Huang and Belongie [51] is data-driven and limited in generalising on unseen styles. Also, simply adjusting the mean and variance of feature statistics makes it hard to synthesise complicated style patterns with rich details and local structures.

A more recent work by Li et al. [59] attempts to exploit a series of feature transformations to transfer arbitrary artistic style in a style learning free manner. Similar to [51], Li et al. use the first few layers of pre-trained VGG as the encoder and train the corresponding decoder. But they replace the AdaIN layer [51] in between the encoder and decoder with a pair of whitening and colouring transformations (WCT): $I = Dec(\ \text{WCT}(\mathcal{F}(I_c), \mathcal{F}(I_s))\ )$. Their algorithm is built on the observation that the whitening transformation can remove the style related information and preserve the structure of content. Therefore, receiving content activations $\mathcal{F}(I_c)$ from the encoder, whitening transformation can filter the original style out of the input content image and return a filtered representation with only content information. Then, by applying colouring transformation, the style patterns contained in $\mathcal{F}(I_s)$ are incorporated into the filtered content representation, and the stylised result I can be obtained by decoding the transformed features. They also extend this single-level stylisation to multi-level stylisation to further improve visual quality.

The algorithm of Li et al. is the first ASPM algorithm to transfer artistic styles in a learning-free manner. Therefore, compared with [51], it does not have the limitation in generalisation capabilities. But the algorithm of Li et al. is still not effective at producing sharp details and fine strokes. The stylisation results will be shown in Section 6. Also, it lacks a consideration in preserving depth information and variations in brush strokes.

5 Improvements and Extensions

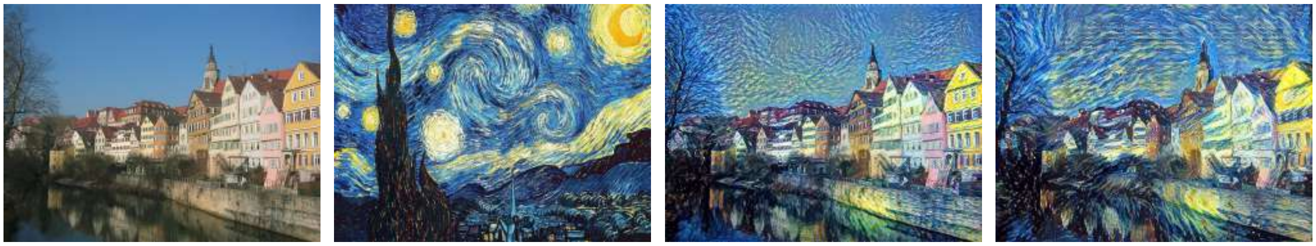
Since the emergence of NST algorithms, there are also some researches devoted to improving current NST algorithms by controlling perceptual factors (e.g., stroke size control, spatial style control, and colour control) (Figure 2, green boxes). Also, all of aforementioned NST methods are designed for general still images. They may not be appropriate for specialised types of images and videos (e.g., doodles, head portraits, and video frames). Thus, a variety of follow-up studies (Figure 2, pink boxes) aim to extend general NST algorithms to these particular types of images and even extend them beyond artistic image style (e.g., audio style).

Controlling Perceptual Factors in Neural Style Transfer. Gatys et al. themselves [60] propose several slight modifications to improve their previous algorithm [4]. They demonstrate a spatial style control strategy to control the style in each region of the content image. Their idea is to define guidance channels for the feature activations for both content and style image. The guidance channel has values in $[0, 1]$ specifying which style should be transferred to which content region, i.e., the content regions where the content guidance channel is 1 should be rendered with the style where the style guidance channel is equal to 1. While for the colour control, the original NST algorithm produces stylised images with the colour distribution of the style image. However, sometimes people prefer a colour-preserving style transfer, i.e., preserving the colour of the content image during style transfer. The corresponding solution is to first transform the style image's colours to match the content image's colours before style transfer, or alternatively perform style transfer only in the luminance channel.

For stroke size control, the problem is much more complex. We show sample results of stroke size control in Figure 3. The discussions of stroke size control strategy need to be split into several cases [61]:

1) IOB-NST with non-high-resolution images: Since current style statistics (e.g., Gram-based and BN-based statistics) are scale-sensitive [61], to achieve different stroke sizes, the solution is simply resizing a given style image to different scales.

2) MOB-NST with non-high-resolution images: One possible solution is to resize the input image to different scales before the forward pass, which inevitably hurts stylisation quality. Another possible solution is to train multiple models with different scales of a style image, which is space and time consuming. Also, the possible solution fails to preserve *stroke consistency* among results with different stroke sizes, i.e., the results vary in stroke orientations, stroke configurations, etc. However, users generally desire to only change



(a) Content (b) Style (c) Small Stroke Size (d) Large Stroke Size

Figure 3: Control the brush stroke size in NST. (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.

the stroke size but not others. To address this problem, Jing et al. [61] propose a stroke controllable PSPM algorithm. The core component of their algorithm is a *StrokePyramid* module, which learns different stroke sizes with adaptive receptive fields. Without trading off quality and speed, their algorithm is the first to exploit one single model to achieve flexible continuous stroke size control while preserving *stroke consistency*, and further achieve spatial stroke size control to produce new artistic effects. Although one can also use ASPM algorithm to control stroke size, ASPM trades off quality and speed. As a result, ASPM is not effective at producing fine strokes and details compared with [61].

3) IOB-NST with high-resolution images: For high-resolution images (e.g., 3000×3000 pixels in [60]), a large stroke size cannot be achieved by simply resizing style image to a large scale. Since only the region in the content image with a receptive field size of VGG can be affected by a neuron in the loss network, there is almost no visual difference between a large and larger brush strokes in a small image region with receptive field size. Gatys et al. [60] tackle this problem by proposing a coarse-to-fine IOB-NST procedure with several steps of downsampling, stylising, upsampling and final stylising.

4) MOB-NST with high-resolution images: Similar to 3), stroke size in stylised result does not vary with style image scale for high-resolution images. The solution is also similar to Gatys et al. 's algorithm in [60], which is a coarse-to-fine stylisation procedure [62]. The idea is to exploit a multimodel, which comprises multiple subnetworks. Each subnetwork receives the upsampled stylised result of the previous subnetwork as the input, and stylises it again with finer strokes.

Another limitation of current NST algorithms is that they do not consider the depth information contained in the image. To address this limitation, the depth preserving NST algorithm [63] is proposed. Their approach is to add a depth loss function based on [47] to measure the depth difference between the content image and the stylised image. The image depth is acquired by applying a single-image depth estimation algorithm (e.g., Chen et al.'s work in [64]).

Semantic Style Transfer. Given a pair of style and content images which are similar in content, the goal of semantic style transfer is to build a semantic correspondence between the style and content, which maps each style region to a corresponding semantically similar content region. Then the style in each style region is transferred to the semantically similar content region.

1) Image-Optimisation-Based Semantic Style Transfer. Since the patch matching scheme naturally meets the requirements of the region-based correspondence, Champandard [65] proposes to build a semantic style transfer algorithm based on the aforementioned patch-based algorithm [46] (Section 4.1.2). Although the result produced by the algorithm of Li and Wand [46] is close to the target of semantic style transfer, [46] does not incorporate an accurate segmentation mask, which sometimes leads to a wrong semantic match. Therefore, Champandard augments an additional semantic channel upon [46], which is a downsampled semantic segmentation map. The segmentation map can be either manually annotated or from a semantic segmentation algorithm [66], [67]. Despite the effectiveness of [65], MRF-based design is not the only choice. Instead of combining MRF prior, Chen and Hsu [68] provide an alternative way for semantic style transfer, which is to exploit masking out process to constrain the spatial correspondence and also a higher order style feature statistic to further improve the result. More recently, Mechrez et al. [69] propose an alternative contextual loss to realise semantic style transfer in a segmentation-free manner.

2) Model-Optimisation-Based Semantic Style Transfer. As before, the efficiency issue is always a big issue. Both [65] and [68] are based on IOB-NST algorithms and therefore leave much room for improvement. Lu et al. [70] speed up the process by optimising the objective function in feature space, instead of in pixel space. More specifically, they propose to do feature reconstruction, instead of image reconstruction as previous algorithms do. This optimisation strategy reduces the computation burden, since the loss does not need to propagate through a deep network. The resulting reconstructed feature is decoded into the final result with a trained decoder. Since the speed of [70] does not reach real-time, there is still big room for further research.

Instance Style Transfer. Instance style transfer is built on instance segmentation and aims to stylise only a single user-specified object within an image. The challenge mainly lies in the transition between a stylised object and non-stylised background. Castillo et al. [71] tackle this problem by adding an extra MRF-based loss to smooth and anti-alias boundary pixels.

Doodle Style Transfer. An interesting extension can be found in [65], which is to exploit NST to transform rough sketches into fine artworks. The method is simply discarding content loss term and using doodles as segmentation map to do semantic style transfer.

Stereoscopic Style Transfer. Driven by the demand of AR/VR, Chen et al. [72] propose a stereoscopic NST algorithm for stereoscopic images. They propose a disparity loss to penalise the bidirectional disparity. Their algorithm is shown to produce more consistent strokes for different views.

Portrait Style Transfer. Current style transfer algorithms are usually not optimised for head portraits. As they do not impose spatial constraints, directly applying these existing algorithms to head portraits will deform facial structures, which is unacceptable for the human visual system. Selim et al. [73] address this problem and extend [4] to head portrait painting transfer. They propose to use the notion of gain maps to constrain spatial configurations, which can preserve the facial structures while transferring the texture of the style image.

Video Style Transfer. NST algorithms for video sequences are substantially proposed shortly after Gatys et al.'s first NST algorithm for still images [4]. Different from still image style transfer, the design of video style transfer algorithm needs to consider the smooth transition between adjacent video frames. Like before, we divide related algorithms into Image-Optimisation-Based and Model-Optimisation-Based Video Style Transfer.

1) Image-Optimisation-Based Online Video Style Transfer. The first video style transfer algorithm is proposed by Ruder et al. [74], [75]. They introduce a temporal consistency loss based on optical flow to penalise the deviations along point trajectories. The optical flow is calculated by using novel optical flow estimation algorithms [76], [77]. As a result, their algorithm eliminates temporal artefacts and produces smooth stylised videos. However, they build their algorithm upon [4] and need several minutes to process a single frame.

2) Model-Optimisation-Based Offline Video Style Transfer. Several follow-up studies are devoted to stylising a given video in real-time. Huang et al. [78] propose to augment Ruder et al.'s temporal consistency loss [74] upon current PSPM algorithm. Given two consecutive frames, the temporal consistency loss is directly computed using two corresponding outputs of style transfer network to encourage pixel-wise consistency, and a corresponding two-frame synergic training strategy is introduced for the computation of temporal consistency loss. Another concurrent work which shares a similar idea with [78] but with an additional exploration of style instability problem can be found in [79]. Different from [78], [79], Chen et al. [80] propose a flow subnetwork to produce feature flow and incorporate optical flow information in feature space. Their algorithm is built on a pre-trained style transfer network (an encoder-decoder pair) and wraps feature activations from the pre-trained stylisation encoder using the obtained feature flow.

Character Style Transfer. Given a style image containing multiple characters, the goal of *Character Style Transfer* is to apply the idea of NST to generate new fonts and text effects. In [81], Atarsaikhan et al. directly apply the algorithm in [4] to font style transfer and achieve visually plausible results. While Yang et al. [82] propose to first characterise style elements and exploit extracted characteristics to guide the generation of text effects. A more recent work [83] designs a conditional GAN model for glyph shape prediction, and also an ornamentation network for colour and texture prediction. By training these two networks jointly, font style transfer can be realised in an end-to-end manner.

Photorealistic Style Transfer. Photorealistic style transfer (also known as colour style transfer) aims to transfer the style of colour distributions. The general idea is to build upon current semantic style transfer but to eliminate distortions and preserve the original structure of the content image.

1) Image-Optimisation-Based Photorealistic Style Transfer. The earliest photorealistic style transfer approach is proposed by Luan et al. [84]. They propose a two-stage optimisation procedure, which is to initialise the optimisation by stylising a given photo with non-photorealistic style transfer algorithm [65] and then penalise image distortions by adding a photorealism regularization. But since Luan et al.'s algorithm is built on the *Image-Optimisation-Based Semantic Style Transfer* method [65], their algorithm is computationally expensive. Similar to [84], another algorithm proposed by Mechrez et al. [85] also adopts a two-stage optimisation procedure. They propose to refine the non-photorealistic stylised result by matching the gradients in the output image to those in the content photo. Compared to [84], the algorithm of Mechrez et al. achieves a faster photorealistic stylisation speed.

2) Model-Optimisation-Based Photorealistic Style Transfer. Li et al. [86] address the efficiency issue of [84] by handling this problem with two steps, the stylisation step and smoothing step. The stylisation step is to apply the NST algorithm in [59] but replace upsampling layers with unpooling layers to produce the stylised result with fewer distortions. Then the smoothing step further eliminates structural artefacts. These two aforementioned algorithms [84], [86] are mainly designed for natural images. Another work in [87] proposes to exploit GAN to transfer the colour from human-designed anime images to sketches. Their algorithm demonstrates a promising application of Photorealistic Style Transfer, which is the automatic image colourisation.

Attribute Style Transfer. Image attributes are generally referred to image colours, textures, etc. Previously, image attribute transfer is accomplished through image analogy [9] in a supervised manner (Section 2). Derived from the idea of patch-based NST [46], Liao et al. [88] propose a deep image analogy to study image analogy in the domain of CNN features. Their algorithm is based on a patch matching technique and realises a weakly supervised image analogy, i.e., their algorithm only needs a single pair of source and target images instead of a large training set.

Fashion Style Transfer. Fashion style transfer receives fashion style image as the target and generates clothing images with desired fashion styles. The challenge of Fashion Style Transfer lies in the preservation of similar design with the basic input clothing while blending desired style patterns. This idea is first proposed by Jiang and Fu [89]. They tackle this problem by proposing a pair of fashion style generator and discriminator.

Audio Style Transfer. In addition to transferring image styles, [90], [91] extend the domain of image style to audio style, and synthesise new sounds by transferring the desired style from a target audio. The study of audio style transfer also follows the route of image style transfer, i.e., *Audio-Optimisation-Based Online Audio Style Transfer* and

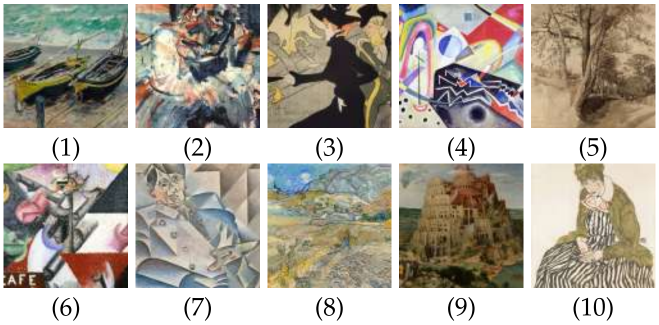


Figure 4: Diversified style images used in our experiment.

Table 1: Detailed information of our style images.

No.	Author	Name & Year
1	Claude Monet	*Three Fishing Boats* (1886)
2	Georges Rouault	*Head of a Clown* (1907)
3	Henri de Toulouse-Lautrec	*Divan Japonais* (1893)
4	Wassily Kandinsky	*White Zig Zags* (1922)
5	John Ruskin	*Trees in a Lane* (1847)
6	Severini Gino	*Ritmo plastico del 14 luglio* (1913)
7	Juan Gris	*Portrait of Pablo Picasso* (1912)
8	Vincent van Gogh	*Landscape at Saint-Rémy* (1889)
9	Pieter Bruegel the Elder	*The Tower of Babel* (1563)
10	Egon Schiele	*Edith with Striped Dress* (1915)

Note: All our style images are in the public domain.

then *Model-Optimisation-Based Offline Audio Style Transfer*. Inspired by image-based IOB-NST, Verma and Smith [90] propose a *Audio-Optimisation-Based Online Audio Style Transfer* algorithm based on online audio optimisation. They start from a noise signal and optimise it iteratively using backpropagation. [91] improves the efficiency by transferring an audio in a feed-forward manner and can produce the result in real-time.

6 Evaluation Methodology

The evaluations of NST algorithms remain an open and important problem in this field. In general, there are two major types of evaluation methodologies that can be employed in the field of NST, i.e., qualitative evaluation and quantitative evaluation. Qualitative evaluation relies on the aesthetic judgements of observers. The evaluation results are related to lots of factors (e.g., age and occupation of participants). While quantitative evaluation focuses on the precise evaluation metrics, which include time complexity, loss variation, etc. In this section, we experimentally compare different NST algorithms both qualitatively and quantitatively.

6.1 Experimental Setup

Evaluation datasets. Totally, there are ten style images and twenty content images used in our experiment.

For style images, we select artworks of diversified styles, as shown in Figure 4. For example, there are impressionism, cubism, abstract, contemporary, futurism, surrealist, and expressionism art. Regarding the mediums, some of these artworks are painted on canvas, while others are painted on cardboard or wool, cotton, polyester, etc. In addition, we also try to cover a range of image characteristics (such as details, contrast, complexity and color distributions), inspired by the works in [92], [93], [95]. More detailed information of our style images are given in Table 1.

For content images, there are already carefully selected and well-described benchmark datasets for evaluating stylisation by Mould and Rosin [92], [93], [95]. Their proposed NPR benchmark called *NPRgeneral* consists of the images that cover a wide range of characteristics (e.g., contrast, texture, edges and meaningful structures) and satisfy lots of criteria. Therefore, we directly use the selected twenty images in their proposed *NPRgeneral* benchmark as our content images.

For the algorithms based on offline model optimisation, MS-COCO dataset [96] is used to perform the training. All the content images are not used in training.

Principles. To maximise the fairness of the comparisons, we also obey the following principles during our experiment:

1) In order to cover every detail in each algorithm, we try to use the provided implementation from their published literatures. To maximise the fairness of comparison especially for speed comparison, for [10], we use a popular torch-based open source code [97], which is also admitted by the authors. In our experiment, except for [32], [53] which are based on TensorFlow, all the other codes are implemented based on Torch 7.

2) Since the visual effect is influenced by the content and style weight, it is difficult to compare results with different degrees of stylisation. Simply giving the same content and style weight is not an optimal solution due to the different ways to calculate losses in each algorithm (e.g., different choices of content and style layers, different loss functions). Therefore, in our experiment, we try our best to balance the content and style weight among different algorithms.

3) We try to use the default parameters (e.g., choice of layers, learning rate, etc) suggested by the authors except for the aforementioned content and style weight. Although the results for some algorithms may be further improved by more careful hyperparameter tuning, we select the authors' default parameters since we hold the point that the *sensitivity for hyperparameters* is also an important implicit criterion for comparison. For example, we cannot say an algorithm is effective if it needs heavy work to tune its parameters for each style.

There are also some other implementation details to be noted. For [47] and [48], we use the instance normalisation strategy proposed in [50], which is not covered in the published papers. Also, we do not consider the diversity loss term (proposed in [50], [55]) for all algorithms, i.e., one pair of content and style images corresponds to one stylised result in our experiment. For Chen and Schmidt's algorithm [57], we use the feed-forward reconstruction to reconstruct the stylised results.

6.2 Qualitative Evaluation

Example stylised results are shown in Figure 5, Figure 7 and Figure 9. More results can be found in the supplementary

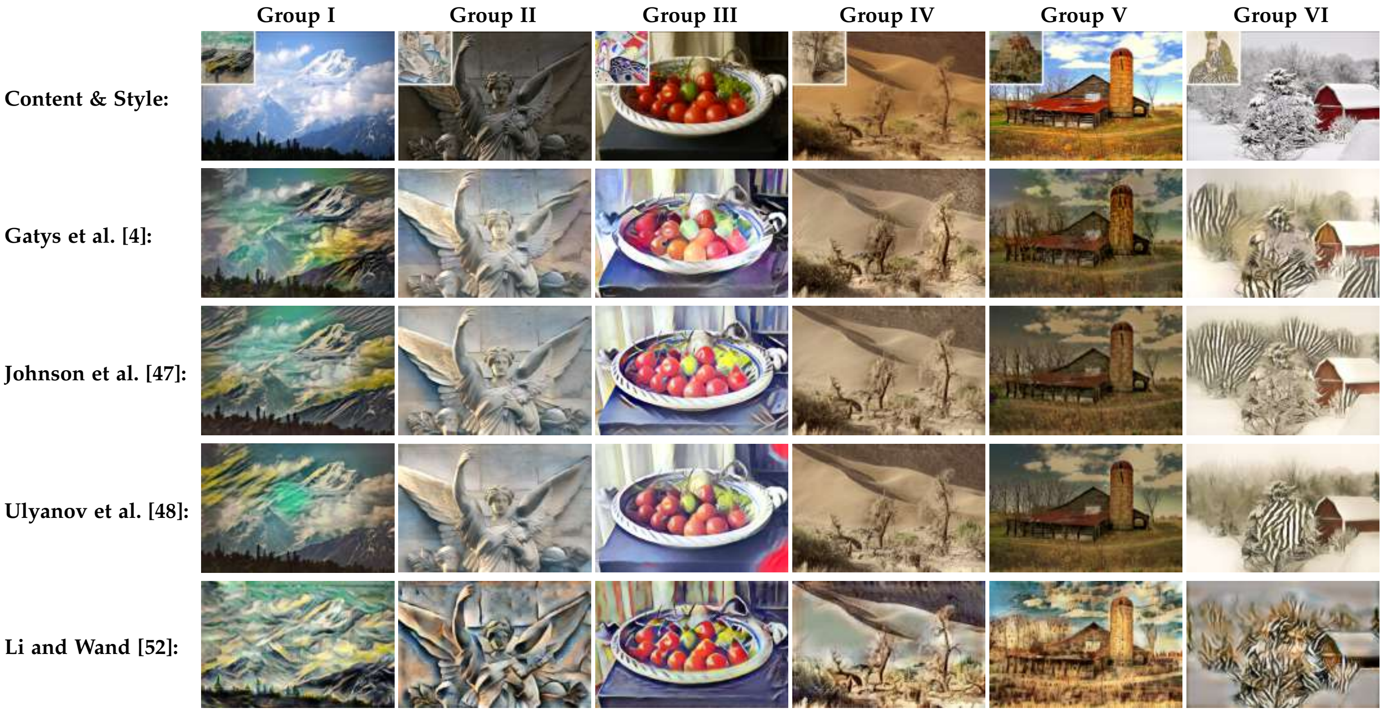


Figure 5: Some example results of **IOB-NST** and **PSPM-MOB-NST** for qualitative evaluation. The content images are from the benchmark dataset proposed by Mould and Rosin [92], [93]. The style images are in the public domain. Detailed information of our style images can be found in Table 1.

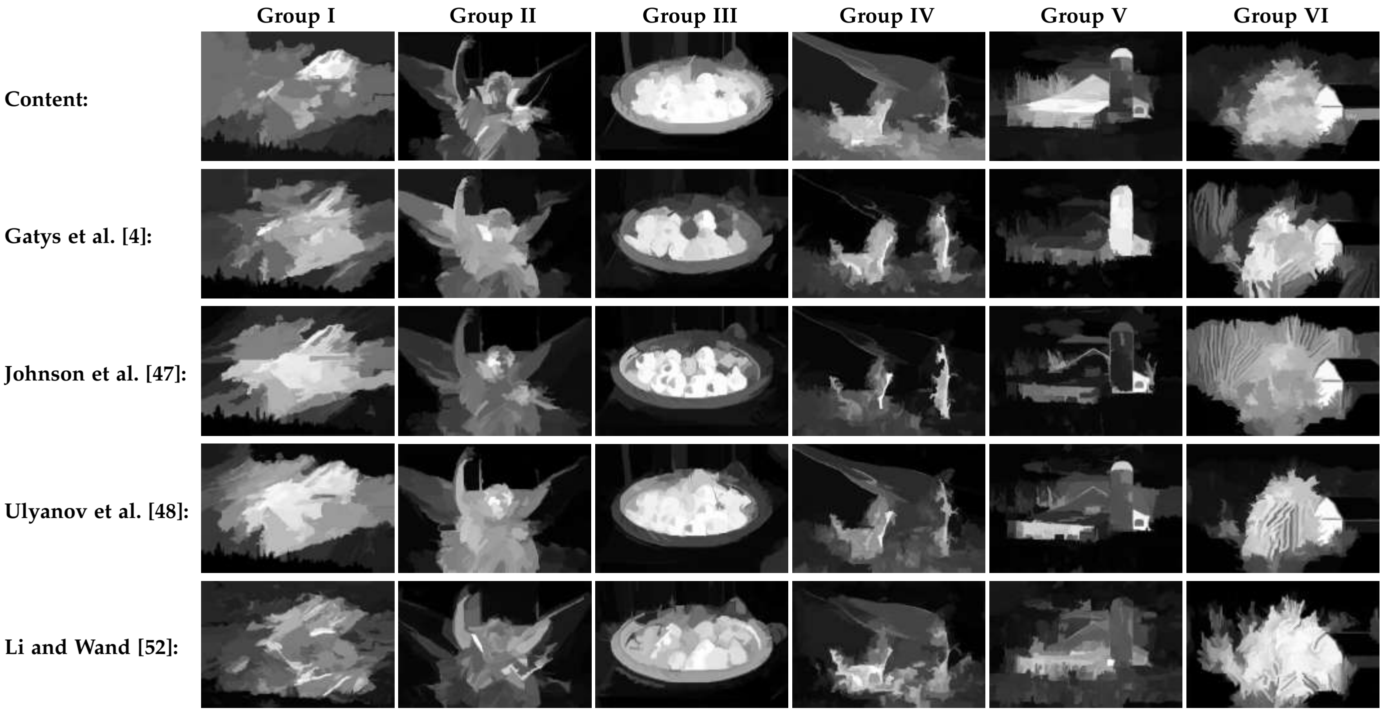


Figure 6: Saliency detection results of **IOB-NST** and **PSPM-MOB-NST**, corresponding to Figure 5. The results are produced by using the discriminative regional feature integration approach proposed by Wang et al. [94].

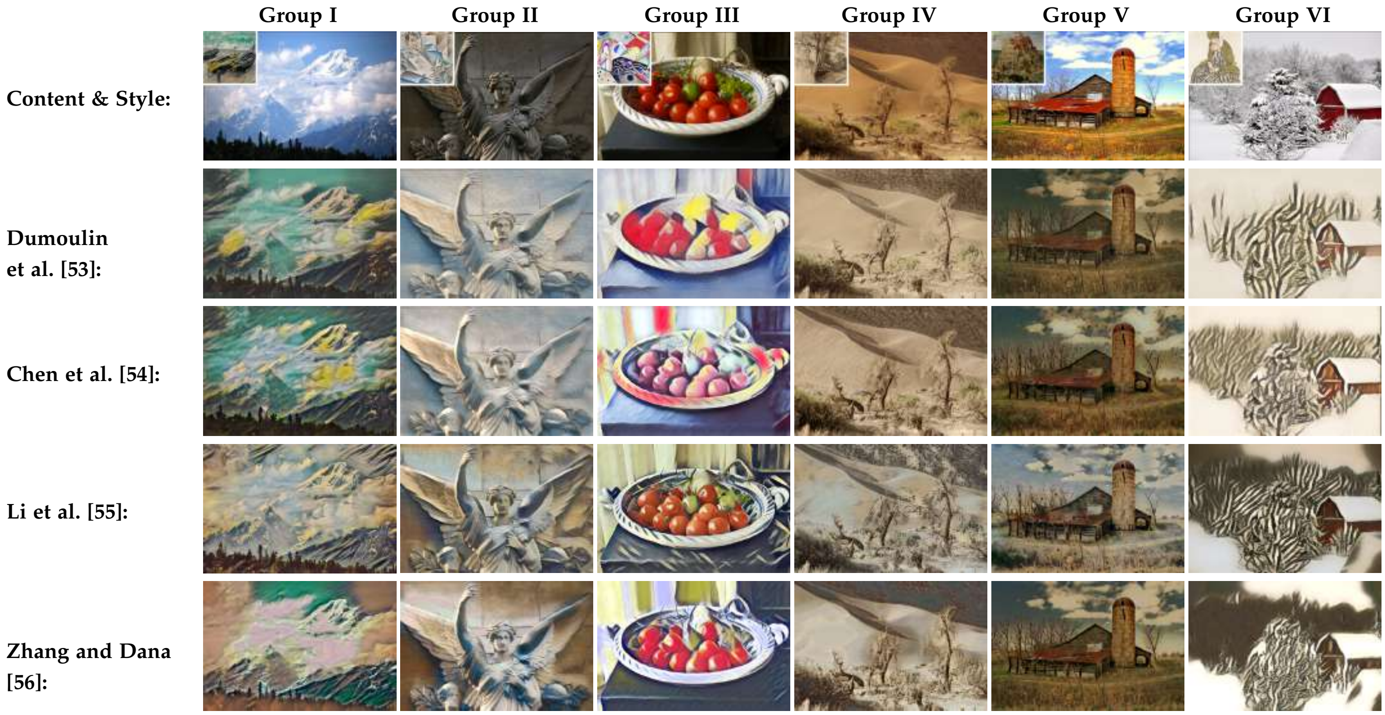


Figure 7: Some example results of **MSPM-MOB-NST** for qualitative evaluation. The content images are from the benchmark dataset proposed by Mould and Rosin [92], [93]. The style images are in the public domain. Detailed information of our style images can be found in Table 1.

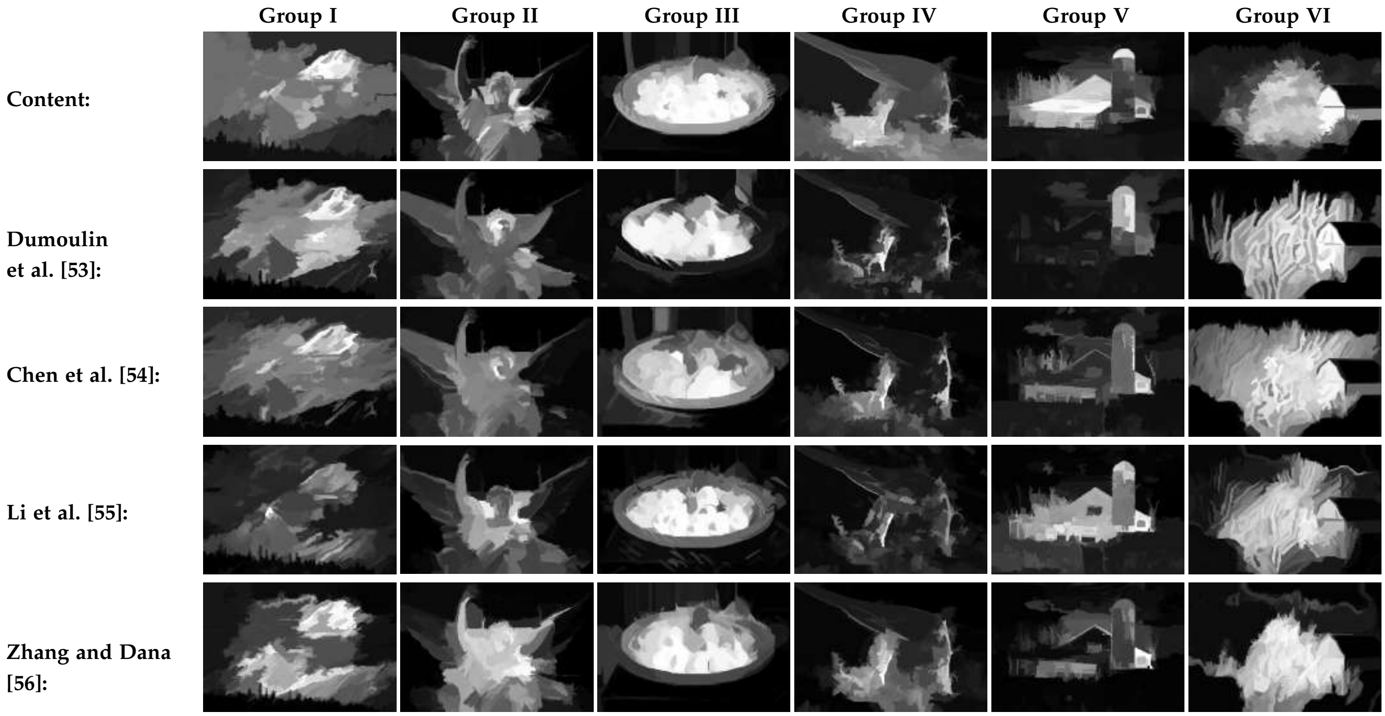


Figure 8: Saliency detection results of **MSPM-MOB-NST**, corresponding to Figure 7. The results are produced by using the discriminative regional feature integration approach proposed by Wang et al. [94].

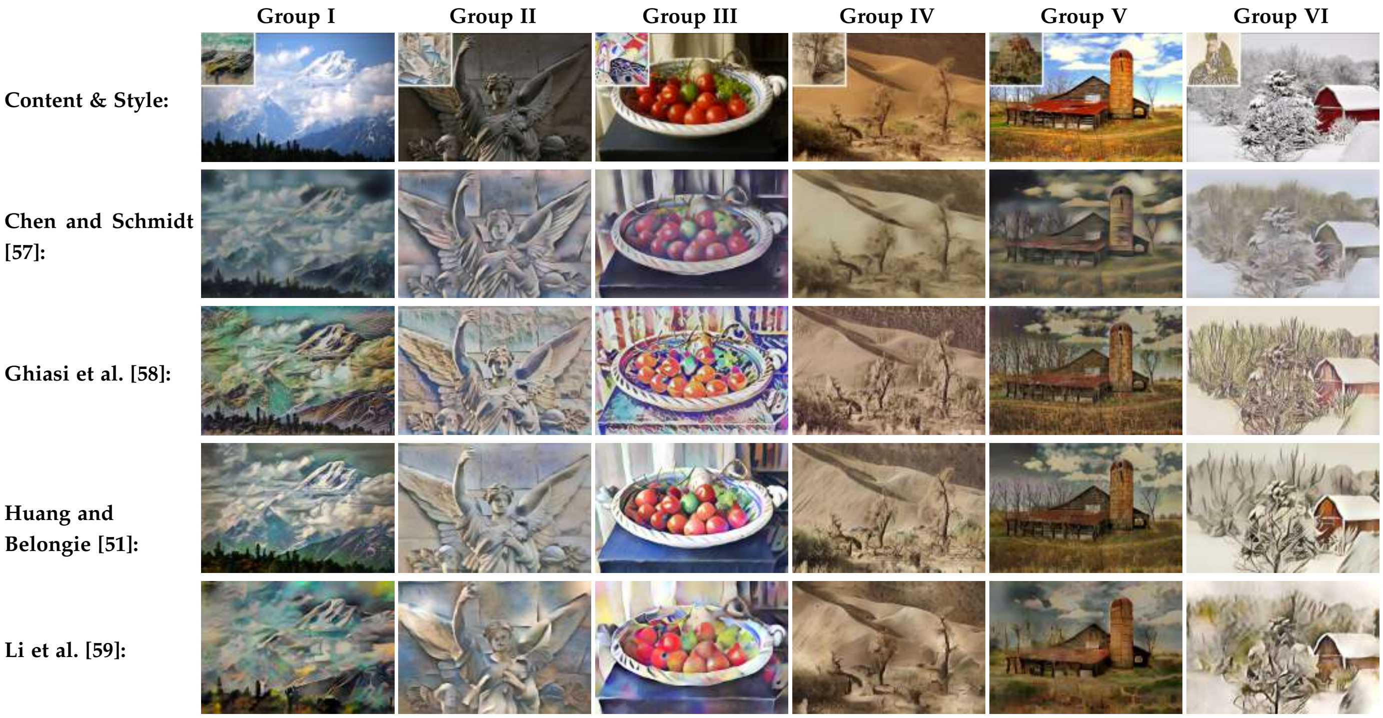


Figure 9: Some example results of **ASPM-MOB-NST** for qualitative evaluation. The content images are from the benchmark dataset proposed by Mould and Rosin [92], [93]. The style images are in the public domain. Detailed information of our style images can be found in Table 1.

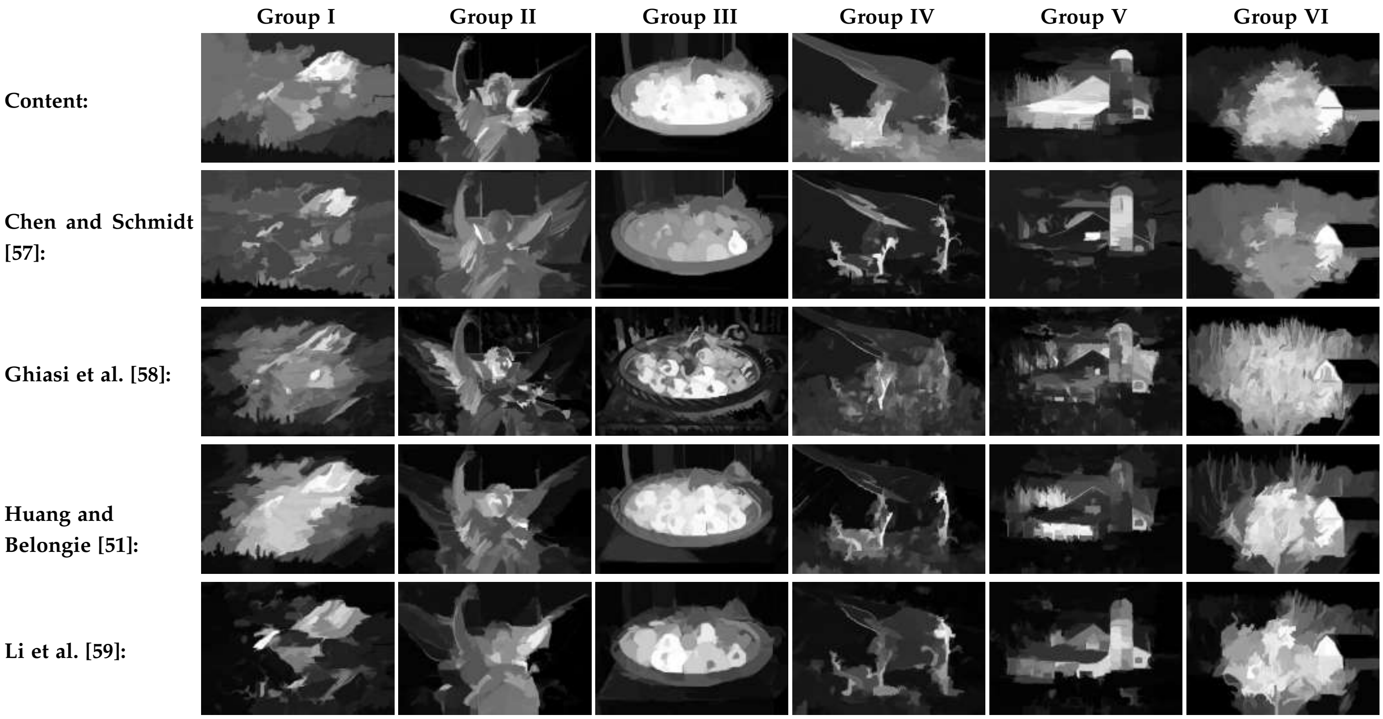


Figure 10: Saliency detection results of **ASPM-MOB-NST**, corresponding to Figure 9. The results are produced by using the discriminative regional feature integration approach proposed by Wang et al. [94].

material[3].

1) Results of IOB-NST. Following the content and style images, Figure 5 contains the results of Gatys et al.'s IOB-NST algorithm based on online image optimisation [4]. The style transfer process is computationally expensive, but in contrast, the results are appealing in visual quality. Therefore, the algorithm of Gatys et al. is usually regarded as the gold-standard method in the community of NST.

2) Results of PSPM-MOB-NST. Figure 5 shows the results of *Per-Style-Per-Model* MOB-NST algorithms (Section 4.2). Each model only fits one style. It can be noticed that the stylised results of Ulyanov et al. [48] and Johnson et al. [47] are somewhat similar. This is not surprising since they share a similar idea and only differ in their detailed network architectures. For the results of Li and Wand [52], the results are sightly less impressive. Since [52] is based on Generative Adversarial Network (GAN), to some extent, the training process is not that stable. But we believe that GAN-based style transfer is a very promising direction, and there are already some other GAN-based works [83], [87], [98] (Section 5) in the field of NST.

3) Results of MSPM-MOB-NST. Figure 7 demonstrates the results of *Multiple-Style-Per-Model* MOB-NST algorithms. Multiple styles are incorporated into a single model. The idea of both Dumoulin et al.'s algorithm [53] and Chen et al.'s algorithm [54] is to tie a small number of parameters to each style. Also, both of them build their algorithm upon the architecture of [47]. Therefore, it is not surprising that their results are visually similar. Although the results of [53], [54] are appealing, their model size will become larger with the increase of the number of learned styles. In contrast, Zhang and Dana's algorithm [56] and Li et al.'s algorithm [55] use a single network with the same trainable network weights for multiple styles. The model size issue is tackled, but there seem to be some interferences among different styles, which slightly influences the stylisation quality.

4) Results of ASPM-MOB-NST. Figure 9 presents the last category of MOB-NST algorithms, namely *Arbitrary-Style-Per-Model* MOB-NST algorithms. Their idea is one-model-for-all. Globally, the results of ASPM are slightly less impressive than other types of algorithms. This is acceptable in that a three-way trade-off between speed, flexibility and quality is common in research. Chen and Schmidt's patch-based algorithm [57] seems to not combine enough style elements into the content image. Their algorithm is based on similar patch swap. When lots of content patches are swapped with style patches that do not contain enough style elements, the target style will not be reflected well. Ghiasi et al.'s algorithm [58] is data-driven and their stylisation quality is very dependent on the varieties of training styles. For the algorithm of Huang and Belongie [51], they propose to match global summary feature statistics and successfully improve the visual quality compared with [57]. However, their algorithm seems not good at handling complex style patterns, and their stylisation quality is still related to the varieties of training styles. The algorithm of Li et al. [59] replaces the training process with a series of transformations. But [59] is not effective at producing sharp details and fine strokes.

Saliency Comparison. NST is an art creation process. As indicated in [3], [38], [39], the definition of *style* is subjective and also very complex, which involves personal preferences, texture compositions as well as the used tools and medium. As a result, it is difficult to define the aesthetic criterion for a stylised artwork. For the same stylised result, different people may have different or even opposite views. Nevertheless, our goal is to compare the results of different NST techniques (shown in Figure 5, Figure 7 and Figure 9) as objectively as possible. Here, we consider comparing saliency maps, as proposed in [63]. The corresponding results are shown in Figure 6, Figure 8 and Figure 10. Saliency maps can demonstrate visually dominant locations in images. Intuitively, a successful style transfer could weaken or enhance the saliency maps in content images, but should not change the integrity and coherence. From Figure 6 (saliency detection results of IOB-NST and PSPM-MOB-NST), it can be noticed that the stylised results of [4], [47], [48] preserve the structures of content images well; however, for [52], it might be harder for an observer to recognise the objects after stylisation. Using similar analytical method, from Figure 8 (saliency detection results of MSPM-MOB-NST), [53] and [54] preserve similar saliency of the original content images since they both tie a small number of parameters to each style. [56] and [55] are also similar regarding the ability to retain the integrity of the original saliency maps, because they both use a single network for all styles. As shown in Figure 10, for the saliency detection results of ASPM-MOB-NST, [58] and [51] perform better than [57] and [59]; however, both [58] and [51] are data-driven methods and their quality depends on the diversity of training styles. In general, it seems that the results of MSPM-MOB-NST preserve better saliency coherence than ASPM-MOB-NST, but a little inferior to IOB-NST and PSPM-MOB-NST.

6.3 Quantitative Evaluation

Regarding the quantitative evaluation, we mainly focus on five evaluation metrics, which are: generating time for a single content image of different sizes; training time for a single model; average loss for content images to measure how well the loss function is minimised; loss variation during training to measure how fast the model converges; style scalability to measure how large the learned style set can be.

1) Stylisation speed. The issue of efficiency is the focus of MOB-NST algorithms. In this subsection, we compare different algorithms quantitatively in terms of the stylisation speed. Table 2 demonstrates the average time to stylise one image with three resolutions using different algorithms. In our experiment, the style images have the same size as the content images. The fifth column in Table 2 represents the number of styles one model of each algorithm can produce. $k(k \in Z^{+})$ denotes that a single model can produce multiple styles, which corresponds to MSPM algorithms. ∞ means a single model works for any style, which corresponds to ASPM algorithms. The numbers reported in Table 2 are obtained by averaging the generating time of 100 images. Note that we do not include the speed of [53], [58] in Table 2

[3] https://www.dropbox.com/s/5xd8iizoigvjcxz/SupplementaryMaterial_neuralStyleReview.pdf?dl=0

Table 2: Average speed comparison of NST algorithms for images of size 256×256 pixels, 512×512 pixels and 1024×1024 pixels (on an NVIDIA Quadro M6000)

Methods	**Time(s)**			**Styles/Model**
	256 × 256	**512 × 512**	**1024 × 1024**	
Gatys et al. [10]	14.32	51.19	200.3	∞
Johnson et al. [47]	0.014	0.045	0.166	1
Ulyanov et al. [48]	0.022	0.047	0.145	1
Li and Wand [52]	0.015	0.055	0.229	1
Zhang and Dana [56]	0.019 (**0.039**)	0.059 (**0.133**)	0.230 (**0.533**)	$k(k \in Z^{+})$
Li et al. [55]	0.017	0.064	0.254	$k(k \in Z^{+})$
Chen and Schmidt [57]	0.123 (**0.130**)	1.495 (**1.520**)	−	∞
Huang and Belongie [51]	0.026 (**0.037**)	0.095 (**0.137**)	0.382 (**0.552**)	∞
Li et al. [59]	0.620	1.139	2.947	∞

Note: The fifth column shows the number of styles that a single model can produce. Time both excludes (out of parenthesis) and includes (in parenthesis) the style encoding process is shown, since [56], [57] and [51] support storing encoded style statistics in advance to further speed up the stylisation process for the same style but different content images. Time of [57] for producing 1024×1024 images is not shown due to the memory limitation. The speed of [53], [58] are similar to [47] since they share similar architecture. We do not redundantly list them in this table.

Table 3: A summary of the advantages and disadvantages of the mentioned algorithms in our experiment.

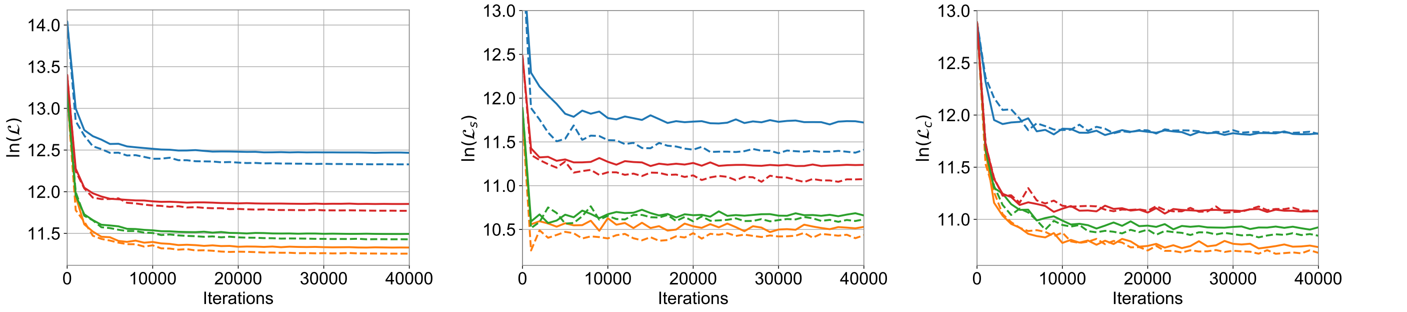
Types	**Methods**	**Pros & Cons**			
		E	**AS**	**LF**	**VQ**
IOB-NST	Gatys et al. [4]	×	√	√	Good and usually regarded as a gold standard.
PSPM- MOB-NST	Ulyanov et al. [47]	√	×	×	The results of [47], [50] are close to [4]. [52] is generally less appealing than [47], [50].
	Johnson et al. [50]	√	×	×	
	Li and Wand [52]	√	×	×	
MSPM- MOB-NST	Dumoulin et al. [53]	√	×	×	The results of [53] and [54] are close to [4], but the model size generally becomes larger with the increase of the number of learned styles. [55], [56] have a fixed model size but there seem to be some interferences among different styles.
	Chen et al. [54]	√	×	×	
	Li et al. [55]	√	×	×	
	Zhang and Dana [56]	√	×	×	
ASPM- MOB-NST	Chen and Schmidt [57]	√	√	×	In general, the results of ASPM are less impressive than other types of NST algorithms. [57] does not combine enough style elements. [51], [58] are generally not effective at producing complex style patterns. [59] is not good at producing sharp details and fine strokes.
	Ghiasi et al. [58]	√	√	×	
	Huang and Belongie [51]	√	√	×	
	Li et al. [59]	√	√	√	

Note: **E**, **AS**, **LF**, and **VQ** represent ***Efficient***, ***Arbitrary Style***, ***Learning-Free***, and ***Visual Quality***, respectively. IOB-NST denotes the category *Image-Optimisation-Based Neural Style Transfer* and MOB-NST represents *Model-Optimisation-Based Neural Style Transfer*.

as their algorithm is to scale and shift parameters based on the algorithm of Johnson et al. [47]. The time required to stylise one image using [32], [53] is very close to [47] under the same setting. For Chen et al.'s algorithm in [54], since their algorithm is protected by patent and they do not make public the detailed architecture design, here we just attach the speed information provided by the authors for reference: On a Pascal Titan X GPU, 256×256: 0.007s; 512×512: 0.024s; 1024×1024: 0.089s. For Chen and Schmidt's algorithm [57], the time for processing a 1024×1024 image is not reported due to the limit of video memory. Swapping patches for two 1024×1024 images needs more than 24 GB video memory and thus, the stylisation process is not practical. We can observe that except for [57], [59], all the other MOB-NST algorithms are capable of stylising even high-resolution content images in real-time. ASPM algorithms are generally slower than PSPM and MSPM, which demonstrates the aforementioned three-way trade-off again.

2) Training time. Another concern is the training time for one single model. The training time of different algorithms is hard to compare as sometimes the model trained with just a few iterations is capable of producing enough visually appealing results. So we just outline our training time of different algorithms (under the same setting) 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. [47], 3 hours for the algorithm of Ulyanov et al. [48], 2 hours for the algorithm of Li and Wand [52], 4 hours for Zhang and Dana [56], and 8 hours for Li et al. [55]. Chen and Schmidt's algorithm [57] and Huang and Belongie's algorithm [51] take much longer (e.g., a couple of days), which is acceptable since a pre-trained model can work for any style. The training time of [58] depends on how large the training style set is. For MSPM algorithms, the training time can be further reduced through incremental learning over a pre-trained model. For example, the algorithm of Chen et al. only needs 8 minutes to incrementally learn a new style, as reported in [54].

3) Loss comparison. One way to evaluate some MOB-NST algorithms which share the same loss function is to compare their loss variation during training, i.e., the training curve comparison. It helps researchers to justify the

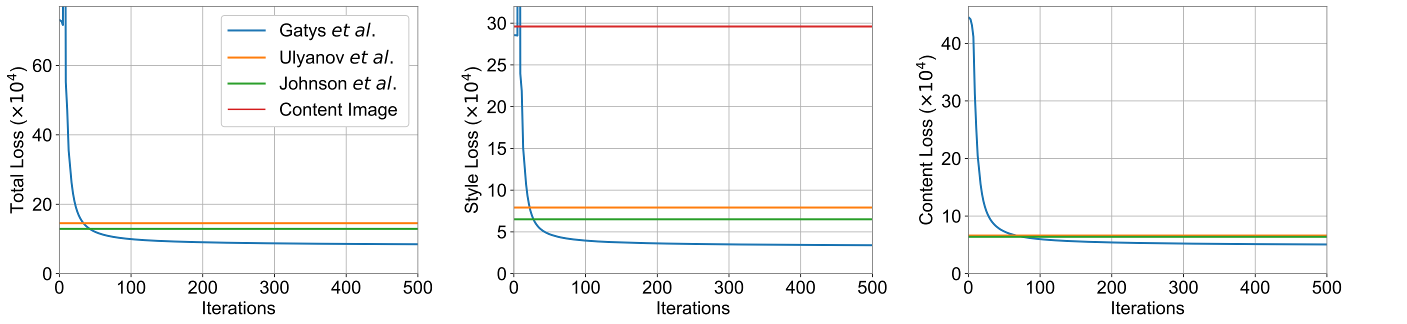


(a) Total Loss Curve

(b) Style Loss Curve

(c) Content Loss Curve

Figure 11: Training curves of total loss, style loss and content loss of different algorithms. Solid curves represent the loss variation of the algorithm of Ulyanov et al. [48], while the dashed curves represent the algorithm of Johnson et al. [47]. Different colours correspond to different randomly selected styles from our style set.



(a) Total Loss

(b) Style Loss

(c) Content Loss

Figure 12: Average total loss, style loss and content loss of different algorithms [4], [47], [48]. The reported numbers are averaged over our set of style and content images.

choice of architecture design by measuring how fast the model converges and how well the same loss function can be minimised. Here we compare training curves of two popular MOB-NST algorithms [47], [48] in Figure 11, since most of the follow-up works are based on their architecture designs. We remove the total variation term and keep the same objective for both two algorithms. Other settings (e.g., loss network, chosen layers) are also kept the same. For the style images, we randomly select four styles from our style set and represent them in different colours in Figure 11. It can be observed that the two algorithms are similar in terms of the convergence speed. Also, both algorithms minimise the content loss well during training, and they mainly differ in the speed of learning the style objective. The algorithm in [47] minimises the style loss better.

Another related criterion is to compare the final loss values of different algorithms over a set of test images. This metric demonstrates how well the same loss function can be minimised by using different algorithms. For a fair comparison, the loss function and other settings are also required to be kept the same. We show the results of one IOB-NST algorithm [4] and two MOB-NST algorithms [47], [48] in Figure 12. The result is consistent with the aforementioned trade-off between speed and quality. Although MOB-NST algorithms are capable of stylising images in real-time, they are not good as IOB-NST algorithms in terms of minimising the same loss function.

4) Style scalability. Scalability is a very important criterion for MSPM algorithms. However, it is very hard to measure since the maximum capabilities of a single model is highly related to the set of particular styles. If most styles have somewhat similar patterns, a single model can produce thousands of styles or even more, since these similar styles share somewhat similar distribution of style feature statistics. In contrast, if the style patterns vary a lot among different style images, the capability of a single model will be much smaller. But it is hard to measure how much these styles differ from each other in style patterns. Therefore, to provide the reader a reference, here we just summarise the authors' attempt for style scalability: the number is 32 for [53], 1000 for both [54] and [55], and 100 for [56].

A summary of the advantages and disadvantages of the mentioned algorithms in this experiment section can be found in Table 3.

7 Applications

Due to the visually plausible stylised results, the research of NST has led to many successful industrial applications and begun to deliver commercial benefits. In this section, we summarise these applications and present some potential usages.

7.1 Social Communication

One reason why NST catches eyes in both academia and industry is its popularity in some social networking sites, e.g., Facebook and Twitter. A recently emerged mobile application named *Prisma* [11] is one of the first industrial applications that provide the NST algorithm as a service. Due to its high stylisation quality, *Prisma* achieved great success and is becoming popular around the world. Some other applications providing the same service appeared one after another and began to deliver commercial benefits, e.g., a web application *Ostagram* [12] requires users to pay for a faster stylisation speed. Under the help of these industrial applications [13], [99], [100], people can create their own art paintings and share their artwork with others on Twitter and Facebook, which is a new form of social communication. There are also some related application papers: [101] introduces an iOS app *Pictory* which combines style transfer techniques with image filtering; [102] further presents the technical implementation details of *Pictory*; [103] demonstrates the design of another GPU-based mobile app *ProsumerFX*.

The application of NST in social communication reinforces the connections between people and also has positive effects on both academia and industry. For academia, when people share their own masterpiece, their comments can help the researchers to further improve the algorithm. Moreover, the application of NST in social communication also drives the advances of other new techniques. For instance, inspired by the real-time requirements of NST for videos, Facebook AI Research (FAIR) first developed a new mobile-embedded deep learning system *Caffe2Go* and then *Caffe2* (now merged with PyTorch), which can run deep neural networks on mobile phones [104]. For industry, the application brings commercial benefits and promotes the economic development.

7.2 User-assisted Creation Tools

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

As a creation tool for painters and designers, NST can make it more convenient for a painter to create an artwork of a particular style, especially when creating computer-made artworks. Moreover, with NST algorithms, it is trivial to produce stylised fashion elements for fashion designers and stylised CAD drawings for architects in a variety of styles, which will be costly when creating them by hand.

7.3 Production Tools for Entertainment Applications

Some entertainment applications such as movies, animations and games are probably the most application forms of NST. For example, creating an animation usually requires 8 to 24 painted frames per second. The production costs will be largely reduced if NST can be applied to automatically stylise a live-action video into an animation style. Similarly, NST 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 NST for production, e.g., Joshi et al. explore the use of NST in redrawing some scenes in a movie named *Come Swim* [105], which indicates the promising potential applications of NST in this field. In [106], Fišer et al. study an illumination-guided style transfer algorithm for stylisation of 3D renderings. They demonstrate how to exploit their algorithm for rendering previews on various geometries, autocomplete shading, and transferring style without a reference 3D model.

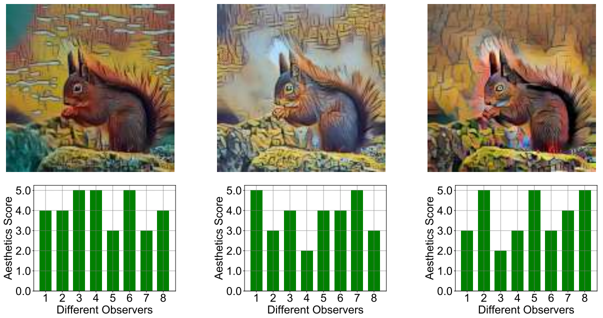


Figure 13: Example of aesthetic preference scores for the outputs of different algorithms given the same style and content.

8 Future Challenges

The advances in the field of NST are inspiring and some algorithms have already found use in industrial applications. Although current algorithms are capable of good performance, there are still several challenges and open issues. In this section, we summarise key challenges within this field of NST and discuss possible strategies on how to deal with them in future works. Since NST is very related to NPR, some critical problems in NPR (summarised in [3], [14], [107], [108], [109], [110]) also remain future challenges for the research of NST. Therefore, we first review some of the major challenges existing in both NPR and NST and then discuss the research questions specialised for the field of NST.

8.1 Evaluation Methodology

Aesthetic evaluation is a critical issue in both NPR and NST. In the field of NPR, the necessity of aesthetic evaluation is explained by many researchers [3], [14], [107], [108], [109], [110], e.g., in [3], Rosin and Collomosse use two chapters to explore this issue. This problem is increasingly critical as the fields of NPR and NST mature. As pointed out in [3], researchers need some reliable criteria to assess the benefits of their proposed approach over the prior art and also a way to evaluate the suitability of one particular approach to one particular scenario. However, most NPR and NST papers evaluate their proposed approach with side-by-side subjective visual comparisons, or through measurements derived from various user studies [59], [111], [112]. For example, to evaluate the proposed universal style transfer algorithm, Li et al. [59] conduct a user study which is to ask participants to vote for their favourite stylised results. We argue that it is not an optimal solution since the results vary a lot with different observers. Inspired by [113], we conduct a simple experiment for user studies with the stylised results of different NST algorithms. In our experiment, each stylised image is rated by 8 different raters (4 males and 4 females) with the same occupation and age. As depicted in Figure 13, given the same stylised result, different observers with the same occupation and age still have quite different ratings. Nevertheless, there is currently no gold standard evaluation method for assessing NPR and NST algorithms. This challenge of aesthetic evaluation will continue to be an open question in both NPR and NST communities, the solution of which might require the collaboration with professional artists and the efforts in the identification of underlying aesthetic principles.

In the field of NST, there is another important issue related to aesthetic evaluation. Currently, there is no standard benchmark image set for evaluating NST algorithms. Different authors typically use their own images for evaluation. In our experiment, we use the carefully selected NPR benchmark image set named *NPRgeneral* [92], [93] as our content images to compare different techniques, which is backed by the comprehensive study in [92], [93]; however, we have to admit that the selection of our style images is far from being a standard NST benchmark style set. Different from NPR, NST algorithms do not have explicit restrictions on the types of style images. Therefore, to compare the style scalability of different NST methods, it is critical to seek a benchmark style set which collectively exhibits a broad range of possible properties, accompanied by a detailed description of adopted principles, numerical measurements of image characteristics as well as a discussion of limitations like the works in [92], [93], [95]. Based on the above discussion, seeking an NST benchmark image set is quite a separate and important research direction, which provides not only a way for researchers to demonstrate the improvement of their proposed approach over the prior art, but also a tool to measure the suitability of one particular NST algorithm to one particular requirement. In addition, as the emergence of several NST extensions (Section 5), it remains another open problem to study the specialised benchmark data set and also the corresponding evaluation criteria for assessing those extended works (e.g., video style transfer, audio style transfer, stereoscopic style transfer, character style transfer and fashion style transfer).

8.2 Interpretable Neural Style Transfer

Another challenging problem is the interpretability of NST algorithms. Like many other CNN-based vision tasks, the process of NST is like a black box, which makes it quite uncontrollable. In this part, we focus on three critical issues related to the interpretability of NST, i.e., interpretable and controllable NST via disentangled representations, normalisation methods associated with NST, and adversarial examples in NST.

Representation disentangling. The goal of representation disentangling is to learn dimension-wise interpretable representations, where some changes in one or more specific dimensions correspond to changes precisely in a single factor of variation while being invariant to other factors [114], [115], [116], [117]. Such representations are useful to a variety of machine learning tasks, e.g., visual concepts learning [118] and transfer learning [119]. For example, in style transfer, if one could learn a representation where the factors of variation (e.g., colour, shape, stroke size, stroke orientation and stroke composition) are precisely disentangled, these factors could then be freely controlled during stylisation. For example, one could change the stroke orientations in a stylised image by simply changing the corresponding dimension in the learned disentangled representation. Towards the goal of disentangled representation, current methods fit into two categories, which are supervised approaches and unsupervised ones. The basic idea of supervised disentangling methods is to exploit annotated data to supervise the mapping between inputs and attributes [120], [121]. Despite their effectiveness, supervised disentangling approaches typically require numbers of training samples. However, in the case of NST, it is quite complicated to model and capture some of those aforementioned factors of variation. For example, it is hard to collect a set of images which have different stroke orientations but exactly the same colour distribution, stroke size and stroke composition. By contrast, unsupervised disentangling methods do not require annotations; however, they usually yield disentangled representations which are dimension-wise uncontrollable and uninterpretable [122], i.e., we could not control what would be encoded in each specific dimension. Based on the above discussion, to acquire disentangled representations in NST, the first issue to be addressed is how to define, model and capture the complicated factors of variation in NST.

Table 4: Normalisation methods in NST.

Paper	Author	Name
[50]	Ulyanov et al.	*Instance Normalisation*
[53]	Dumoulin et al.	*Conditional Instance Normalisation*
[51]	Huang and Belongie	*Adaptive Instance Normalisation*

Normalisation methods. The advances in the field of NST are closely related to the emergence of novel normalisation methods, as shown in Table 4. Some of these normalisation methods also have an influence on a larger vision community beyond style transfer (e.g., image recolourisation [123] and video colour propagation [124]). In this part, we first briefly review these normalisation methods in NST and then discuss the corresponding problem. The first emerged normalisation method in NST is *instance normalisation* (or *contrast normalisation*) proposed by Ulyanov et al. [50]. *Instance normalisation* is equivalent to *batch normalisation* when the batch size is one. It is shown that style transfer network with *instance normalisation* layer converges faster and produces visually better results compared with the network with *batch normalisation* layer. Ulyanov et al. believe that the superior performance of *instance normalisation* results from the fact that *instance normalisation* enables the network to discard contrast information in content images and therefore makes learning simpler. Another explanation proposed by Huang and Belongie [51] is that *instance normalisation* performs a kind of *style normalisation* by normalising feature statistics (i.e., the mean and variance). With *instance normalisation*, the style of each individual image could be directly normalised to the target style. As a result, the rest of the network only needs to take care of the content loss, making the objective easier to learn. Based on *instance normalisation*, Dumoulin et al. [53] further propose *conditional instance normalisation*, which is to scale and shift parameters in *instance normalisation* layers (shown in Equation (8)). Following the interpretation proposed by Huang and Belongie, by using different affine parameters, the feature statistics could be normalised to different values. Correspondingly, the style of each individual sample could be normalised to different styles. Furthermore, in [51], Huang and Belongie propose *adaptive instance normalisation* to adaptively instance normalise content feature by the style feature statistics (shown in Equation (9)). In this way, they believe that the style of an individual image could be normalised to arbitrary styles. Despite the superior performance achieved by *instance normalisation*, *conditional instance normalisation* and *adaptive instance normalisation*, the reason behind their success still remains unclear. Although Ulyanov et al. [50] and Huang and Belongie [51] propose their own hypothesis based on pixel space and feature space respectively, there is a lack of theoretical proof for their proposed theories. In addition, their proposed theories are also built on other hypothesises, e.g., Huang and Belongie propose their interpretation based on the observation by Li et al. [42]: channel-wise feature statistics, namely mean and variance, could represent styles. However, it remains uncertain why feature statistics could represent the style, or even whether the feature statistics could represent all styles, which relates back to the interpretability of style representations.

Adversarial examples. Several studies have shown that deep classification networks are easily fooled by *adversarial examples* [125], [126], which are generated by applying perturbations to input images (e.g., Figure 14(c)). Previous studies on adversarial examples mainly focus on deep classification networks. However, as shown in Figure 14, we find that adversarial examples also exist in generative style transfer networks. In Figure 14(d), one can hardly recognise the content, which is originally contained in Figure 14(c). It reveals the difference between generative networks and the human vision system. The perturbed image is still recognisable to humans but leads to a different result for generative style transfer networks. However, it remains unclear why some perturbations could make such a difference, and whether some similar noised images uploaded by the user could still be stylised into the desired style. Interpreting and understanding adversarial examples in NST could help to avoid some failure cases in stylisation.

8.3 Three-way Trade-off in Neural Style Transfer

In the field of NST, there is a three-way trade-off between speed, flexibility and quality. IOB-NST achieves superior performance in quality but is computationally expensive. PSPM-MOB-NST achieves real-time stylisation; however, PSPM-MOB-NST needs to train a separate network for each style, which is not flexible. MSPM-MOB-NST improves the flexibility by incorporating multiple styles into one single model, but it still needs to pre-train a network for a set of target styles. Although ASPM-MOB-NST algorithms successfully transfer arbitrary styles, they are not that satisfying in perceptual quality and speed. The quality of data-driven ASPM quite relies on the diversity of training styles. However, one can hardly cover every style due to the great diversity of artworks. Image transformation based ASPM algorithm transfers arbitrary styles in a learning-free manner, but it is behind others in speed. Another related issue is the problem of hyperparameter tuning. To produce the most visually appealing results, it remains uncertain how to set the value of content and style weights, how to choose layers for computing content and style loss, which optimiser to use and how to set the value of learning rate. Currently, researchers empirically set these hyperparameters; however, one set of hyperparameters does not necessarily work for any style and it is tedious to manually tune these parameters for each combination of content and style images. One of the keys for this problem is a better understanding of the optimisation procedure in NST. A deep understanding of optimisation procedure would help understand how to find the local minima that lead to a high quality.

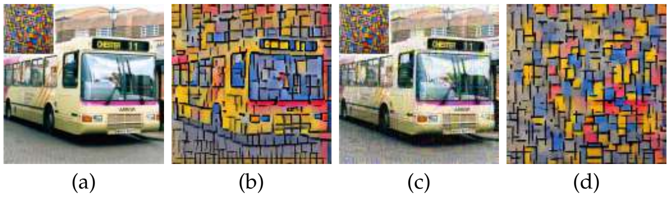


Figure 14: Adversarial example for NST: (a) is the original content and style image pair and (b) is the stylised result of (a) with [47]; (c) is the generated adversarial example and (d) is the stylised result of (c) with the same model as (b).

9 Discussions and Conclusions

Over the past several years, NST has continued to become an inspiring research area, motivated by both scientific challenges and industrial demands. A considerable amount of researches have been conducted in the field of NST. Key advances in this field are summarised in Figure 2. A summary of the corresponding style transfer loss functions can be found in Table 5. NST is quite a fast-paced area, and we are looking forwarding to more exciting works devoted to advancing the development of this field.

During the period of preparing this review, we are also delighted to find that related researches on NST also bring new inspirations for other areas [127], [128], [129], [130], [131] and accelerate the development of a wider vision community. For the area of *Image Reconstruction*, inspired by NST, Ulyanov et al. [127] propose a novel deep image prior, which replaces the manually-designed total variation regulariser in [33] with a randomly initialised deep neural network. Given a task-dependent loss function $\mathcal{L}$, an image I_o and a fixed uniform noise z as inputs, their algorithm can be formulated as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(g_{\theta^*}(z), I_o),\ I^* = g_{\theta^*}(z). \tag{10}$$

One can easily notice that Equation (10) is very similar to Equation (7). The process in [127] is equivalent with

Table 5: An overview of major style transfer loss functions.

Paper	Loss	Description
Gatys et al. [4]	*Gram Loss*	The first proposed style loss based on Gram-based style representations.
Johnson et al. [47]	*Perceptual Loss*	Widely adopted content loss based on perceptual similarity.
Berger and Memisevic [32]	*Transformed Gram Loss*	Computing *Gram Loss* over horizontally and vertically translated feature representations. More effective at modelling style with symmetric properties, compared with *Gram Loss*.
Li et al. [55]	*Mean-substraction Gram Loss*	Subtracting the mean of feature representations before computing *Gram Loss*. Eliminating large discrepancy in scale. Effective at multi-style transfer with one single network.
Zhang and Dana [56]	*Multi-scale Gram Loss*	Computing *Gram Loss* over multi-scale feature representations. Eliminating a few artefacts.
Li et al. [42]	*MMD Loss with Different Kernels*	*Gram Loss* is equivalent to *MMD Loss with Second Order Polynomial Kernel*. *MMD Loss with Linear Kernel* is capable of comparable quality with *Gram Loss*, but with lower computational complexity.
Li et al. [42]	*BN Loss*	Achieving comparable quality with *Gram Loss*, but conceptually clearer in theory.
Risser et al. [44]	*Histogram Loss*	Matching the entire histogram of feature representations. Eliminating instability artefacts, compared with single *Gram Loss*.
Li et al. [45]	*Laplacian Loss*	Eliminating distorted structures and irregular artefacts.
Li and Wand [46]	*MRF Loss*	More effective when the content and style are similar in shape and perspective, compared with *Gram Loss*.
Champandard [65]	*Semantic Loss*	Incorporating a segmentation mask over *MRF Loss*. Enabling a more accurate semantic match.
Li and Wand [52]	*Adversarial Loss*	Computed based on PatchGAN. Utilising contextual correspondence between patches. More effective at preserving coherent textures in complex images, compared with *Gram Loss*.
Jing et al. [61]	*Stroke Loss*	Achieving continuous stroke size control while preserving stroke consistency.
Wang et al. [62]	*Hierarchical Loss*	Enabling a coarse-to-fine stylisation procedure. Capable of producing large but also subtle strokes for high-resolution content images.
Liu et al. [63]	*Depth Loss*	Preserving depth maps of content images. Effective at retaining spatial layout and structure of content images, compared with single *Gram Loss*.
Ruder et al. [74]	*Temporal Consistency Loss*	Designed for video style transfer. Penalising the deviations along point trajectories based on optical flow. Capable of maintaining temporal consistency among stylised video frames.
Chen et al. [72]	*Disparity Loss*	Designed for stereoscopic style transfer. Penalising bidirectional disparity. Capable of consistent strokes for different views.

the training process of MOB-NST when there is only one available image in the training set, but replacing I_c with z and $\mathcal{L}_{total}$ with $\mathcal{L}$. In other words, g in [127] is trained to overfit one single sample. Inspired by NST, Upchurch et al. [128] propose a deep feature interpolation technique and provide a new baseline for the area of *Image Transformation* (e.g., face aging and smiling). Upon the procedure of IOB-NST algorithm [4], they add an extra step which is interpolating in the VGG feature space. In this way, their algorithm successfully changes image contents in a learning-free manner. Another field closely related to NST is *Face Photo-sketch Synthesis*. For example, [132] exploits style transfer to generate shadings and textures for final face sketches. Similarly, for the area of *Face Swapping*, the idea of MOB-NST algorithm [48] can be directly applied to build a feed-forward *Face-Swap* algorithm [133]. NST also provides a new way for *Domain Adaption*, as is validated in the work of Atapour-Abarghouei and Breckon [131]. They apply style transfer technique to translate images from different domains so as to improve the generalisation capabilities of their *Monocular Depth Estimation* model.

Despite the great progress in recent years, the area of NST is far from a mature state. Currently, the first stage of NST is to refine and optimise recent NST algorithms, aiming to perfectly imitate varieties of styles. This stage involves two technical directions. The first one is to reduce failure cases and improve stylised quality on a wider variety of style and content images. Although there is not an explicit restriction on the type of styles, NST does have styles it is particularly good at and also some certain styles it is weak in. For example, NST typically performs well in producing irregular style elements (e.g., paintings), as demonstrated in many NST papers [4], [47], [53], [59]; however, for some styles with regular elements such as low-poly styles [134], [135] and pixelator styles [136], NST generally produces distorted and irregular results due to the property of CNN-based image reconstruction. For content images, previous NST papers usually use natural images as content to demonstrate their proposed algorithms; however, given abstract images (e.g., sketches and cartoons) as input content, NST typically does not combine enough style elements to match the content [137], since a pre-trained classification network could not extract proper image content from these abstract images. The other technical direction of the first stage lies in deriving more extensions from general NST algorithms. For example, as the emergence of 3D vision techniques, it is promising to study 3D surface stylisation, which is to directly optimise and produce 3D objects for both photorealistic and non-photorealistic stylisation. After moving beyond the first stage, a further trend of NST is to not just imitate human-created art with NST techniques, but rather to create a new form of AI-created art under the guidance of underlying aesthetic principles. The first step towards this direction has been taken, i.e., using current NST methods [53], [54], [62] to combine different styles. For example, in [62], Wang et al. successfully utilise their proposed algorithm to produce a new style which fuses the coarse texture distortions of one style with the fine brush strokes of another style image.

References

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

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

[3] P. Rosin and J. Collomosse, *Image and video-based artistic stylisation*. Springer Science & Business Media, 2012, vol. 42.

[4] 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.

[5] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 341–346.

[6] I. Drori, D. Cohen-Or, and H. Yeshurun, "Example-based style synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2003, pp. II–143.

[7] O. Frigo, N. Sabater, J. Delon, and P. Hellier, "Split and match: Example-based adaptive patch sampling for unsupervised style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 553–561.

[8] M. Elad and P. Milanfar, "Style transfer via texture synthesis," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2338–2351, 2017.

[9] 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.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *ArXiv e-prints*, Aug. 2015.

[11] I. Prisma Labs, "Prisma: Turn memories into art using artificial intelligence," 2016. [Online]. Available: http://prisma-ai.com

[12] "Ostagram," 2016. [Online]. Available: http://ostagram.ru

[13] A. J. Champandard, "Deep forger: Paint photos in the style of famous artists," 2015. [Online]. Available: http://deepforger.com

[14] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, "State of the 'art': A taxonomy of artistic stylization techniques for images and video," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 5, pp. 866–885, 2013.

[15] A. Semmo, T. Isenberg, and J. Döllner, "Neural style transfer: A paradigm shift for image-based artistic rendering?" in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2017, pp. 5:1–5:13.

[16] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998, pp. 453–460.

[17] A. Kolliopoulos, "Image segmentation for stylized non-photorealistic rendering and animation," Ph.D. dissertation, University of Toronto, 2005.

[18] B. Gooch, G. Coombe, and P. Shirley, "Artistic vision: painterly rendering using computer vision techniques," in *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*. ACM, 2002, pp. 83–ff.

[19] Y.-Z. Song, P. L. Rosin, P. M. Hall, and J. Collomosse, "Arty shapes," in *Proceedings of the Fourth Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*. Eurographics Association, 2008, pp. 65–72.

[20] M. Zhao and S.-C. Zhu, "Portrait painting using active templates," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2011, pp. 117–124.

[21] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," in *ACM Transactions On Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 1221–1226.

[22] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 1998, pp. 839–846.

[23] B. Gooch, E. Reinhard, and A. Gooch, "Human facial illustrations: Creation and psychophysical evaluation," *ACM Transactions on Graphics*, vol. 23, no. 1, pp. 27–44, 2004.

[24] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the art in example-based texture synthesis," in *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, 2009, pp. 93–117.

[25] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2. IEEE, 1999, pp. 1033–1038.

[26] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 479–488.

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

[28] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 229–238.

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

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

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] G. Berger and R. Memisevic, "Incorporating long-range consistency in cnn-based texture generation," in *International Conference on Learning Representations*, 2017.

[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*, 2015, pp. 5188–5196.

[34] ——, "Visualizing deep convolutional neural networks using natural pre-images," *International Journal of Computer Vision*, vol. 120, no. 3, pp. 233–255, 2016.

[35] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4829–4837.

[36] ——, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 658–666.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[38] X. Xie, F. Tian, and H. S. Seah, "Feature guided texture synthesis (fgts) for artistic style transfer," in *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*. ACM, 2007, pp. 44–49.

[39] M. Ashikhmin, "Fast texture transfer," *IEEE Computer Graphics and Applications*, no. 4, pp. 38–43, 2003.

[40] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," 2015. [Online]. Available: https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 2230–2236. [Online]. Available: https://doi.org/10.24963/ijcai.2017/310

[43] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

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

[45] S. Li, X. Xu, L. Nie, and T.-S. Chua, "Laplacian-steered neural style transfer," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1716–1724.

[46] 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.

[47] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016, pp. 694–711.

[48] 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*, 2016, pp. 1349–1357.

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

[50] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6924–6932.

[51] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[52] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*, 2016, pp. 702–716.

[53] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *International Conference on Learning Representations*, 2017.

[54] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stylebank: An explicit representation for neural image style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1897–1906.

[55] Y. Li, F. Chen, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Diversified texture synthesis with feed-forward networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3920–3928.

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

[57] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," in *Proceedings of the NIPS Workshop on Constructive Machine Learning*, 2016.

[58] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, "Exploring the structure of a real-time, arbitrary neural artistic stylization network," in *Proceedings of the British Machine Vision Conference*, 2017.

[59] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Advances in Neural Information Processing Systems*, 2017, pp. 385–395.

[60] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3985–3993.

[61] Y. Jing, Y. Liu, Y. Yang, Z. Feng, Y. Yu, D. Tao, and M. Song, "Stroke controllable fast style transfer with adaptive receptive fields," in *European Conference on Computer Vision*, 2018.

[62] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5239–5247.

[63] X.-C. Liu, M.-M. Cheng, Y.-K. Lai, and P. L. Rosin, "Depth-aware neural style transfer," in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, 2017, pp. 4:1–4:10.

[64] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Advances in Neural Information Processing Systems*, 2016, pp. 730–738.

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

[66] J. Ye, Z. Feng, Y. Jing, and M. Song, "Finer-net: Cascaded human parsing with hierarchical granularity," in *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 2018, pp. 1–6.

[67] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2018.

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

[69] R. Mechrez, I. Talmi, and L. Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," in *European Conference on Computer Vision*, 2018.

[70] M. Lu, H. Zhao, A. Yao, F. Xu, Y. Chen, and L. Zhang, "Decoder network over lightweight reconstructed feature for fast semantic style transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2469–2477.

[71] 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," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 1348–1352.

[72] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stereoscopic neural style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[73] A. Selim, M. Elgharib, and L. Doyle, "Painting style transfer for head portraits using convolutional neural networks," *ACM Transactions on Graphics*, vol. 35, no. 4, p. 129, 2016.

[74] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *German Conference on Pattern Recognition*, 2016, pp. 26–36.

[75] ——, "Artistic style transfer for videos and spherical images," *International Journal of Computer Vision*, 2018.

[76] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 1385–1392.

[77] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1164–1172.

[78] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu, "Real-time neural style transfer for videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 783–791.

[79] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei, "Characterizing and improving stability in neural style transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4067–4076.

[80] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, "Coherent online video style transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1105–1114.

[81] G. Atarsaikhan, B. K. Iwana, A. Narusawa, K. Yanai, and S. Uchida, "Neural font style transfer," in *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, vol. 5. IEEE, 2017, pp. 51–56.

[82] S. Yang, J. Liu, Z. Lian, and Z. Guo, "Awesome typography: Statistics-based text effects transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7464–7473.

[83] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content gan for few-shot font style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[84] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 6997–7005.

[85] R. Mechrez, E. Shechtman, and L. Zelnik-Manor, "Photorealistic style transfer with screened poisson equation," in *Proceedings of the British Machine Vision Conference*, 2017.

[86] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," in *European Conference on Computer Vision*, 2018.

[87] L. Zhang, Y. Ji, and X. Lin, "Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan," in *Proceedings of the Asian Conference on Pattern Recognition*, 2017.

[88] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 120, 2017.

[89] S. Jiang and Y. Fu, "Fashion style generator," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 3721–3727.

[90] P. Verma and J. O. Smith, "Neural style transfer for audio spectograms," in *Proceedings of the NIPS Workshop on Machine Learning for Creativity and Design*, 2017.

[91] P. K. Mital, "Time domain neural audio style transfer," in *Proceedings of the NIPS Workshop on Machine Learning for Creativity and Design*, 2018.

[92] D. Mould and P. L. Rosin, "A benchmark image set for evaluating stylization," in *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*. Eurographics Association, 2016, pp. 11–20.

[93] ——, "Developing and applying a benchmark for evaluating image stylization," *Computers & Graphics*, vol. 67, pp. 58–76, 2017.

[94] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach." *International Journal of Computer Vision*, vol. 123, no. 2, 2017.

[95] P. L. Rosin, D. Mould, I. Berger, J. Collomosse, Y.-K. Lai, C. Li, H. Li, A. Shamir, M. Wand, T. Wang, and H. Winnemöller, "Benchmarking non-photorealistic rendering of portraits," in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2017, p. 11.

[96] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[97] J. Johnson, "neural-style," https://github.com/jcjohnson/neural-style, 2015.

[98] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2223–2232.

[99] "DeepArt," 2016. [Online]. Available: https://deepart.io/

[100] R. Sreeraman, "Neuralstyler: Turn your videos/photos/gif into art," 2016. [Online]. Available: http://neuralstyler.com/

[101] A. Semmo, M. Trapp, J. Döllner, and M. Klingbeil, "Pictory: Combining neural style transfer and image filtering," in *ACM SIGGRAPH 2017 Appy Hour*. ACM, 2017, pp. 5:1–5:2.

[102] S. Pasewaldt, A. Semmo, M. Klingbeil, and J. Döllner, "Pictory - neural style transfer and editing with coreml," in *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. ACM, 2017, pp. 12:1–12:2.

[103] T. Dürschmid, M. Söchting, A. Semmo, M. Trapp, and J. Döllner, "Prosumerfx: Mobile design of image stylization components," in *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. ACM, 2017, pp. 1:1–1:8.

[104] Y. Jia and P. Vajda, "Delivering real-time ai in the palm of your hand," 2016. [Online]. Available: https://code.facebook.com/posts/196146247499076/delivering-real-time-ai-in-the-palm-of-your-hand

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

[106] J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, and D. Sỳkora, "Stylit: illumination-guided example-based stylization of 3d renderings," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 92, 2016.

[107] D. H. Salesin, "Non-photorealistic animation & rendering: 7 grand challenges," *Keynote talk at NPAR*, 2002.

[108] A. A. Gooch, J. Long, L. Ji, A. Estey, and B. S. Gooch, "Viewing progress in non-photorealistic rendering through heinlein's lens," in *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2010, pp. 165–171.

[109] D. DeCarlo and M. Stone, "Visual explanations," in *Proceedings of the 8th international symposium on non-photorealistic animation and rendering*. ACM, 2010, pp. 173–178.

[110] A. Hertzmann, "Non-photorealistic rendering and the science of art," in *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2010, pp. 147–157.

[111] D. Mould, "Authorial subjective evaluation of non-photorealistic images," in *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering*. ACM, 2014, pp. 49–56.

[112] T. Isenberg, P. Neumann, S. Carpendale, M. C. Sousa, and J. A. Jorge, "Non-photorealistic rendering in context: an observational study," in *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*. ACM, 2006, pp. 115–126.

[113] J. Ren, X. Shen, Z. Lin, R. Mech, and D. J. Foran, "Personalized image aesthetics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 638–647.

[114] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[115] H. Kim and A. Mnih, "Disentangling by factorising," in *International Conference on Machine Learning*, 2018.

[116] Z. Feng, X. Wang, C. Ke, A. Zeng, D. Tao, and M. Song, "Dual swap disentangling," in *Advances in neural information processing systems*, 2018.

[117] Z. Feng, Z. Yu, Y. Yang, Y. Jing, J. Jiang, and M. Song, "Interpretable partitioned embedding for customized multi-item fashion outfit composition," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ACM, 2018, pp. 143–151.

[118] I. Higgins, N. Sonnerat, L. Matthey, A. Pal, C. P. Burgess, M. Botvinick, D. Hassabis, and A. Lerchner, "Scan: learning abstract hierarchical compositional visual concepts," in *International Conference on Learning Representations*, 2018.

[119] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, 2017.

[120] D. Bouchacourt, R. Tomioka, and S. Nowozin, "Multi-level variational autoencoder: Learning disentangled representations from grouped observations," in *AAAI Conference on Artificial Intelligence*, 2018.

[121] C. Wang, C. Wang, C. Xu, and D. Tao, "Tag disentangled generative adversarial networks for object image re-rendering," in *International Joint Conference on Artificial Intelligence*, 2017.

[122] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2017.

[123] J. Cho, S. Yun, K. Lee, and J. Y. Choi, "Palettenet: Image recolorization with given color palette," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 62–70.

[124] S. Meyer, V. Cornillère, A. Djelouah, C. Schroers, and M. Gross, "Deep video color propagation," in *Proceedings of the British Machine Vision Conference*, 2018.

[125] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[126] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[127] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[128] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7064–7073.

[129] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Transactions on Graphics (Proc. of Siggraph 2018)*, 2018.

[130] Q. Fan, D. Chen, L. Yuan, G. Hua, N. Yu, and B. Chen, "Decouple learning for parameterized image operators," in *European Conference on Computer Vision*, 2018.

[131] A. Atapour-Abarghouei and T. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[132] C. Chen, X. Tan, and K.-Y. K. Wong, "Face sketch synthesis with style transfer using pyramid column feature," in *IEEE Winter Conference on Applications of Computer Vision*. Lake Tahoe, USA, 2018.

[133] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3677–3685.

[134] W. Zhang, S. Xiao, and X. Shi, "Low-poly style image and video processing," in *Systems, Signals and Image Processing (IWSSIP), 2015 International Conference on*. IEEE, 2015, pp. 97–100.

[135] M. Gai and G. Wang, "Artistic low poly rendering for images," *The visual computer*, vol. 32, no. 4, pp. 491–500, 2016.

[136] T. Gerstner, D. DeCarlo, M. Alexa, A. Finkelstein, Y. Gingold, and A. Nealen, "Pixelated image abstraction," in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. Eurographics Association, 2012, pp. 29–36.

[137] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," *arXiv preprint arXiv:1804.04732*, 2018.