

# Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/158470/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Zhu, Xuanying, Lin, Mugang, Wen, Kunhui, Zhao, Huihuang and Sun, Xianfang  
ORCID: <https://orcid.org/0000-0002-6114-0766> 2023. Deep deformable artistic font style transfer. *Electronics* 12 (7) , 1561. 10.3390/electronics12071561 file

Publishers page: <https://doi.org/10.3390/electronics12071561>  
<<https://doi.org/10.3390/electronics12071561>>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.


This version is being made available in accordance with publisher policies.

See

<http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



# Deep Deformable Artistic Font Style Transfer

Xuanying Zhu <sup>1</sup>, Mugang Lin <sup>1,2,\*</sup> , Kunhui Wen <sup>1</sup>, Huihuang Zhao <sup>1,2</sup> and Xianfang Sun <sup>3</sup> <sup>1</sup> College of Computer Science and Technology, Hengyang Normal University, Hengyang 421002, China<sup>2</sup> Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang 421002, China<sup>3</sup> School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK

\* Correspondence: mglin@hynu.edu.cn

**Abstract:** The essence of font style transfer is to move the style features of an image into a font while maintaining the font's glyph structure. At present, generative adversarial networks based on convolutional neural networks play an important role in font style generation. However, traditional convolutional neural networks that recognize font images suffer from poor adaptability to unknown image changes, weak generalization abilities, and poor texture feature extractions. When the glyph structure is very complex, stylized font images cannot be effectively recognized. In this paper, a deep deformable style transfer network is proposed for artistic font style transfer, which can adjust the degree of font deformation according to the style and realize the multiscale artistic style transfer of text. The new model consists of a sketch module for learning glyph mapping, a glyph module for learning style features, and a transfer module for a fusion of style textures. In the glyph module, the Deform-Resblock encoder is designed to extract glyph features, in which a deformable convolution is introduced and the size of the residual module is changed to achieve a fusion of feature information at different scales, preserve the font structure better, and enhance the controllability of text deformation. Therefore, our network has greater control over text, processes image feature information better, and can produce more exquisite artistic fonts.

**Keywords:** style transfer; generative adversarial networks; deformable convolutional networks; artistic font generation



**Citation:** Zhu, X.; Lin, M.; Wen, K.; Zhao, H.; Sun, X. Deep Deformable Artistic Font Style Transfer. *Electronics* **2023**, *12*, 1561. <https://doi.org/10.3390/electronics12071561>

Academic Editor: Donghyeon Cho

Received: 27 February 2023

Revised: 19 March 2023

Accepted: 24 March 2023

Published: 26 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

An artistic font is a beautifully deformed font based on traditional fonts [1] from an artistic and decorative interpretation according to the meaning, character shape, and structural features of the texts. Because of their beautiful and interesting eye-catching characteristics, artistic fonts are widely used in propaganda, advertising, trademarks, and other scenarios and are becoming increasingly popular among the public. A traditional artistic font is designed by professional font designers, so its effect is influenced by the professional level of the designers and other factors. In recent years, with the advent and development of machine learning technology [2], people have applied deep learning methods to artistic font generation to achieve better results.

Currently, the majority of image style transfer methods are based on convolutional neural networks (CNNs). These methods adjust a noisy random image by using an optimization function so that the generated image maintains the content of a normal image while keeping part of the style of the original image. Since artistic fonts can be viewed as beautiful images, image style transfer methods can also be applied to artistic font style transfers. However, the key to artistic font generation is to synthesize text texture and add colorful texture information to the target text. Compared with image style transfer methods, artistic font style transfer methods need to extract the edge features of texts more accurately to maintain the integrity of the font structure in the stylization process. As CNNs adopt a fixed shape of convolution kernels and lack internal mechanisms

to adaptively change the shape of convolution kernels, it is difficult to extend them to new tasks with unknown, complex geometric transformations. For example, for visual recognition with fine localization, different locations need to correspond to different scales or perceptual field sizes appropriate to them while the fixed convolutional kernels limit CNNs to satisfying this requirement. For artistic font style transfer methods based on CNNs, both structural disjunction and stroke overlap will occur in the case of a complex glyph structure. In addition, some conventional CNNs will consider some background features as edge features of the text during feature extraction, which leads to the addition of noise points to the image in the feature extraction process of the glyph, resulting in a double shadow and style spillover in the style transfer process. These problems directly lead to stylized font images not being accurately identified.

Shape-matching GAN [3] is an effective model that can realize multiscale deformation of artistic fonts. The model encoder consists of a general CNN and a controllable module. The original controllable module is composed of the same double-branch network of two layers of convolution, the convolution kernel of each convolution layer is 3, and the receptive field of the obtained feature map is 5. The deep network with residuals at different depths performs better than the shallow network, but the higher the number of layers is, the more overfitting will occur. In addition, features will be lost in the convolution process, and a larger receptive field can better ensure the integrity of the information. In summary, the inherent nature of the general convolutions and the small receptive field of the controllable module prevent shape-matching GAN from recognizing complex fonts, resulting in unclear image glyphs and style overflow.

By the above analysis, there are two challenges for artistic font style transfer methods based on CNNs to improve their performances: (1) how to accurately extract the edge features of texts to provide integral glyphs for generating artistic font; (2) how to eliminate double shadow and style overflow caused by noise. In this paper, a novel artistic font generation network is proposed. To address the first challenge, an encoder for glyph generation is designed which introduces a deformable convolution [4,5] that can freely change the receptive field by adjusting the offset of sampling locations, thus improving its ability to the geometric variations of texts and making it learn more complete information of glyphs. Aiming at the second issue, the difference in adjacent pixel values is calculated as a smoothing loss, and the smoothness of image edges is maintained by reducing the loss.

The contributions of this paper are summarized as follows:

(1) A deep deformable artistic font style transfer network (DAF) is proposed which consists of a Sketch module for learning glyph mapping, a Glyph module for learning style features, and a Transfer module for a fusion of style textures. In the Glyph module, a Deform-Resblock encoder (DR encoder) is designed to extract glyph features, in which a dilation convolution and a deformable convolution are used to change the perceptual field so that the encoder focuses on information about more critical features. The deformable convolution can also help better integrate feature information at different scales to ensure that the generated glyphs maintain their complete font structure.

(2) Ghosting is eliminated and the image is smoothed by introducing a smoothing loss function that reduces the difference in the value of adjacent pixels in the image.

(3) Comparing the proposed model with four current advanced artistic font style transfer methods, the experimental results show that the proposed model is effective and has better performance.

The remainder of this paper is organized as follows. Section 2 reviews related work involving deformable convolutional networks, image style transfer, and font style transfer. In Section 3, the proposed DAF model is then proposed with a detailed description. To evaluate the performance of our model, a series of experiments are conducted in Section 4. Finally, we summarize this paper in Section 5.

## 2. Related Work

### 2.1. Deformable Convolutional Networks

Research on CNNs [6,7] dates back to the neocognitron model proposed by Japanese scientist Kunihiko Fukushima [8] in 1980. It is the first neural network to use convolution and downsampling, and it is also the prototype of convolutional neural networks. In 1989, Yann LeCun [9,10] constructed a CNN for computer vision problems, which was one of the first CNNs, i.e., the original version of LeNet. It uses convolutional layers and pooling layers for the first time, and achieves remarkable accuracy in handwritten character recognition tasks. As LeNet continued to be studied and its subsequent variants defined the basic structure of modern CNNs, its success drew attention to the application of CNNs [11]. Recently, CNNs [12] have achieved significant success in visual recognition tasks, but CNNs are limited to modeling large unknown transformations and lack internal mechanisms to handle geometric transformations. They have difficulty handling a finely localized visual recognition of objects of different scales or deformations. Deformable convolutional networks [4,5] overcome these limitations and shortcomings by introducing a deformable convolution module and a deformable RoI pooling module to improve the transform modeling capabilities of CNNs. In the deformable module, the grid sampling positions of the standard convolution are shifted by 2D offsets learned by an extra convolutional layer. Deformable RoI pooling adds 2D offsets to each bin position in the previous RoI pooling. Thus, the sampling and pooling of a deformable convolutional network can vary with the object's different structure so that it can adjust its feature representation according to the object's configuration. Currently, deformable convolutional networks are widely used in fields such as image processing [13–15], complex vision [16,17], pattern recognition [18,19], and other fields [20,21], where they show powerful performance.

### 2.2. Image Style Transfer

Image style transfer is the migration of a style image so that the input image has the style characteristics of the style image. In 2015, Gatys [22] proposed neural style transfer to facilitate image style transfer. However, neural style transfer has some drawbacks. For example, the network must be trained in each migration, which is very slow and cannot achieve real-time migration, and the style migrations on photos may be distorted. To address these problems, Johnson [23] proposed a fast neural style transfer method to train one network for each style image so that only one forward process is needed to obtain the generated image in a test with a given content image. Luan et al. [24] proposed photo style transfer, which solved the photo distortion problem by improving the loss function. Generative adversarial networks (GANs) [25] are neural networks designed to solve the problem of generative modeling in which the generative model learns to capture the statistical distribution of training data and synthesizes samples from the learned distribution. Consequently, GANs have become a prevalent method for image style transfer [26,27]. Conditional GANs (CGANs) [28] introduce image-to-image translation and a structured loss function to make networks not only learn the mapping from an input image to an output image, but also learn the loss function of training this mapping. This feature makes CGANs suitable for image generation problems. Because there are no content and style constraints on CGANs, their output results are more similar to artistic creations, and the effect is significantly improved compared to the other image generation models. In recent years, GAN has become a hot research direction, and more variant models have been proposed, such as CycleGAN [29], Wasserstein GAN (WGAN) [30], deep convolutional GAN (DCGAN) [31], and shape-matching GAN [3]. Compared with other methods, GANs produce richer artistic effects for image style transfer.

### 2.3. Font Style Transfer

Font style transfer [32], the process of extracting artistic features from images of a given style and integrating artistic characters into text images, is a long-standing research problem. Font synthesis is the process of translating a font from one domain to a font



from another domain, and the key to this process is predicting the shapes of the glyphs. Unlike font synthesis, font style transfer is a challenging problem of transferring the color and texture of artistic styles to new glyphs. The BAIR Lab at Berkeley collaborated with Adobe to design a multi-content GAN for font style transfer [33]. First, they developed a new decorative network to predict the color and texture of the final glyphs. Then, Yang et al. [34] researched dynamic artistic text style migration with glyph style degree control and proposed a novel bidirectional shape-matching framework for font style migration. They introduced a scale-aware shape-matching GAN to learn glyph style shape mapping, model the style shape features at multiple scales simultaneously, transfer them to target glyphs, and generate high-quality and controllable artistic text. Subsequently, Zhang et al. proposed a font effect generation model [35] based on pyramid-style features based on Yang's work, using morphological operations to improve the transfer effect. Recently, a diverse transformation network for text style transformation [36] has been proposed, which can generate multiple styles of text images in a single model, allowing all styles to be effectively trained on the network.

### 3. Artistic Font Generation Network

Image style transfer migrates the style of an image to another image. Unlike image style transfer, font style transfer migrates the style of an image into the text of another image. Thus, if the image style transfer method is copied, the structural characteristics of the text will be destroyed. Yang et al. [3] studied fast and controllable artistic text style transfer in terms of font deformation and proposed a shape-matching GAN for text style transfer. However, by repeatedly testing the shape-matching GAN model, we found that is unable to extract clear font features for fonts with complex strokes, which leads to problems such as stroke adhesion, fuzzy edges, and severe deformation of the font. Currently, these problems are addressed by preprocessing the input image, but this method is time-consuming and difficult to implement. To overcome these limitations, in the section, we propose a deep deformable artistic font style transfer network. The key features of this network are the design of the DR encoder to learn font features and extract more image information, and the introduction of a smoothing loss to preserve key edge details of the font images. Thus, the network is better able to extract features, control font deformation, and maintain the structure of complex glyphs.

#### 3.1. Overall Network Architecture

The overall network architecture consists of three main components: (1) a Sketch module for learning glyph mapping, (2) a Glyph module for learning style features to generate deformed font, and (3) a Transfer module for style texture fusion. The network architecture is shown in Figure 1.

As shown in Figure 1a, the network training process is divided into three modules. They are the Sketch module, Glyph module, and Transfer module. In the Sketch module, we first process the style image and turn it into a mask, which can be easily obtained using image editing tools. We call the mask structure mapping  $X$  of the style image and then use  $X$  and the style image as the input of the sketch module  $G_B$ . The sketch module  $G_B$  is composed of a smoothing block, a transformation block, and a smoothing loss function. The smoothing block is used to smooth the input image and the transformation block is used to map the smoothed style image back to the text field. Therefore, the edge of the style image can learn the edge features of the text image and realize structural transformation. We first smooth the input style image mask  $X$  by sketch model  $G_B$  to weaken the edges. We introduce a new loss function called the smooth loss to maintain the smoothness of the image so that the font can better learn the features of the style image that we provide. We transform the style image into different degrees of deformation by adjusting the parameter  $l$  ( $l \in [0, 1]$ ). After deformation, the mask is generated. In the Glyph module, we train the  $G_S$  network. By clipping mask, the training pairs of sketch shapes with different smoothness can be obtained. The training pair is fed into the glyph network  $G_S$ , which is trained to

map it to the original  $X$  so that it can characterize the shape features of  $X$  and transfer these features to the target text. Thus, it can increase data diversity and force the model to learn more robust features, hence effectively improving the generalization ability of the model. Through the  $G_S$  network, the font learns the style structure features and obtains the deformed font mask. In the Transfer module, we train the network  $G_T$  which is similar to training  $G_S$ . It is necessary to randomly cut a style image mask  $X$  and a style image to form a training pair as the input of  $G_T$ . The network  $G_T$  is trained to perform texture rendering instead. Style migration is performed on the input image to allow the deformed font to have the style features of the style image.

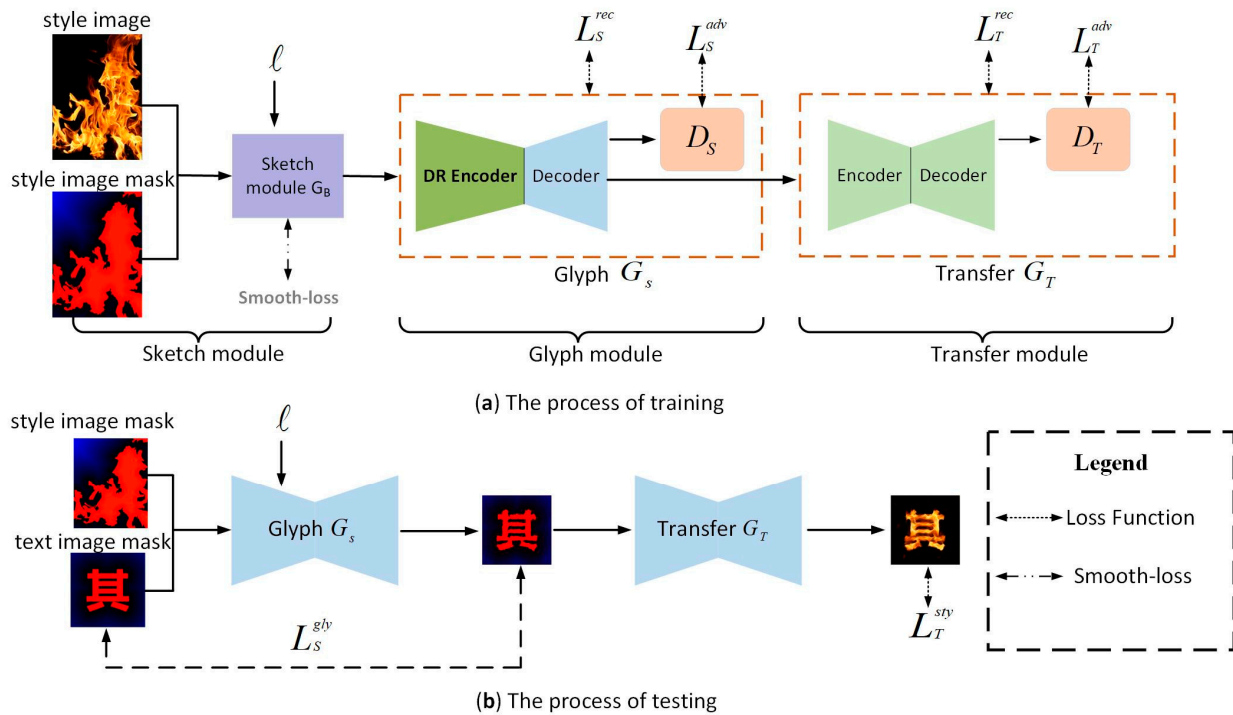


Figure 1. Architecture of the deep deformable artistic font style transfer network.

Figure 1b shows the network testing process.  $G_S$  learns the structural features of style images through training. By inputting text mask images and style mask images, text images can learn the corresponding style features and generate deformed text mask images. The deformed text mask is input into  $G_T$  for style texture migration to obtain the final result.

### 3.2. Glyph Networks ( $G_S$ )

The generator encoder of generative adversarial networks is generally a convolutional neural network, which consists of a convolutional layer, a pooling layer, and a batch normalization layer. The DR encoder is redesigned as shown in Figure 2. We first fill the input feature map repeatedly, filling the feature map to a specific size, and then use dilation convolution to expand the receptive field of the network on the feature map. Second, we downsample the feature map twice and shift the target features through deformable convolution to obtain more accurate edge features. Finally, the feature map is fed into the controllable deep residual network and linearly superimposed, and the corresponding feature map is output. By continuously learning and constantly adjusting the size of the convolutional layers to obtain the most suitable depth for this network, the texture generation network retains as many complex font structure features as possible. Considering that pooling degrades the performance of the generative model, the encoder uses stepwise convolution for reduced sampling. In addition, we use transposed convolution for feature upsampling to avoid checkerboard artifacts.

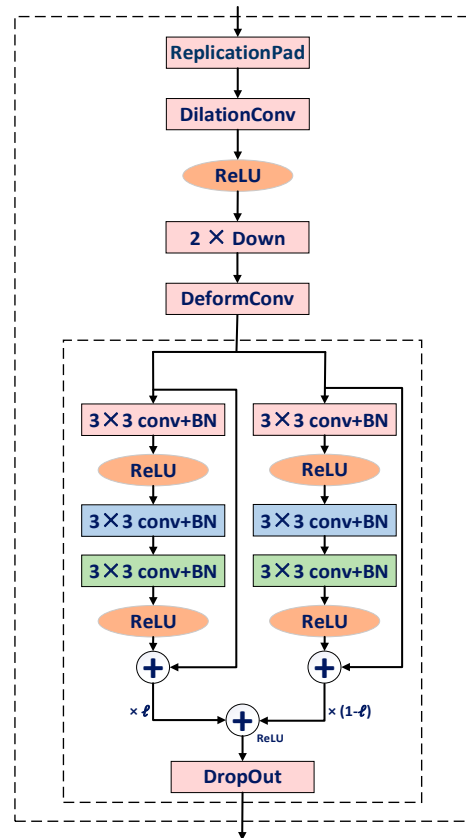


Figure 2. DR encoder structure diagram.

The glyph generation network generator consists of an encoder and a decoder. The encoder is crucial in the glyph network, which determines whether the feature fusion process can maintain the glyph structure. To allow the network to effectively recognize font details, we design the structure of the DR encoder, as shown in Figure 2. The glyph generation network extracts the desired text and style features with the DR encoder and optimizes the training process by learning a large number of samples. After the training process, the network has learnt the corresponding stylistic features and can directly perform stylizations to generate font masks with stylistic features, which significantly reduces the time and space complexity compared with other networks, making the application of style transformation possible.

The structure of the DR encoder is improved mainly by designing the residual module size and introducing deformable convolution. The convolutional layer in the encoder is responsible for acquiring the local image features. The field of perception is fixed by the size of the convolution kernel during the computation of ordinary perception convolutions. We can expand the field of perception only by changing the size of the convolutional kernel or increasing the number of convolutional layers, which inevitably increases the number of parameters and computations of the network model and affects model efficiency. Therefore, we use the dilated convolutional layer instead of the normal convolutional layer to expand the corresponding field of perception without changing the size of the convolutional kernel to increase the network attention to include more features and obtain more detailed information. In CNNs, we calculate the size of the perceptual field by Equation (1):

$$g_n = g_{n-1} + (k_n - 1) * \prod_{i=1}^{n-1} S_i \tag{1}$$

where  $g$  is the receptive field layer,  $n$  is the number of layers,  $S_i$  is the step size of the  $i$ -th layer convolution or pooling, and  $k$  is the size of the convolution kernel which is based on Equation (1) to make the receptive field grow exponentially.

The dilated convolution has a hyperparameter dilation rate  $r$ , which represents the interval of the convolution kernel, the dilation rate of the standard convolution is 1. We calculate  $r$  through Equation (2).

$$r = 2^{\log_2 rate + 2} - 1 \tag{2}$$

In our calculation formula, *rate* defaults to 1. Dilated convolution increases the field of perception of the convolution kernel while keeping the number of parameters constant, so that each convolution output contains a larger range of information, allowing us to better detect feature targets and capture contextual information. However, there is a limitation of convolution for complex font cavities, where too large a perceptual field blurs detailed features when there are more strokes in the font. Therefore, to compensate for the dilated convolution insufficiency, we introduce deformable convolution in the encoder and use additional offsets to increase the spatial sampling position in the module so that our convolutional layer can automatically adjust the scale or perceptual field to obtain the best image.

In addition, the adjustment of the direction vector of the convolution kernel is added to the traditional convolution to shift the morphology of the convolution kernel closer to the feature object. The convolution process is shown in Figure 3.

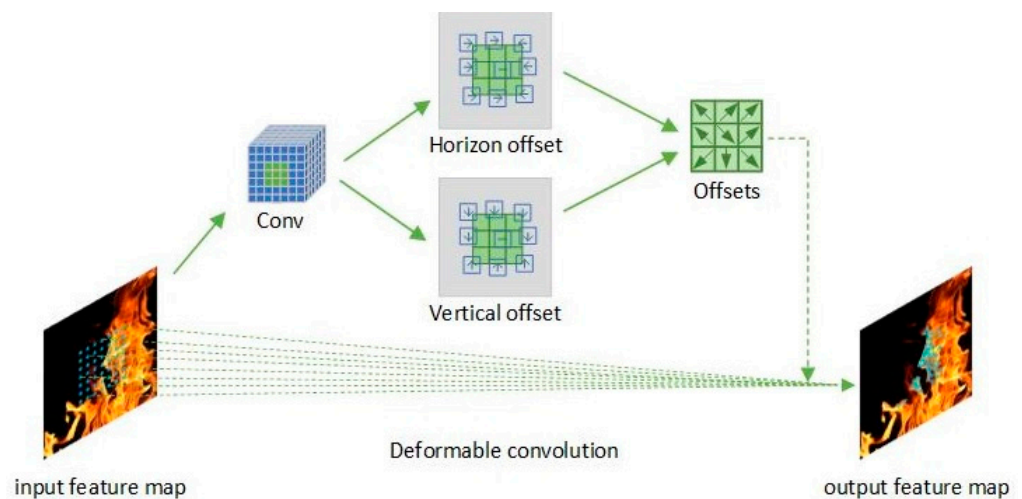


Figure 3. Schematic diagram of the joined deformable convolutional network.

The CNNs can extract the feature maps, use the feature maps as input and apply another convolutional layer to them. In Figure 3, there is an additional convolutional layer to learn the offset and to share the input feature maps. The purpose of this layer is to obtain the offset of the convolutional deformation; we use Equation (3), an interpolation algorithm is used to learn the offset, which is learned by backpropagation. The difference in deformable ConvNets is that they perform dense spatial transformations in a simple, efficient, deep, and end-to-end manner. The deformable convolution introduces an offset  $\Delta P_n$  for each point, which is generated from the input feature map with another convolution, usually a fractional number.  $P_n$  is each offset of  $P_0$  in the range of the convolution kernel in Equation (4) and is represented as follows.

$$y(P_0) = \sum_{P_n \in R} w(P_n) \cdot x(P_0 + P_n + \Delta P_n) \tag{3}$$

$$x(P) = \sum_q \max(0, 1 - |q_x - p_x|) \cdot \max(0, 1 - |q_y - p_y|) \cdot x(q) \tag{4}$$



In the subsequent ablation experiments, it is also verified that the introduction of deformable convolution can better extract the features of text images and style images in the encoder, and improve the control of font deformation. This innovation is the key to solving the problem that the feature extraction of complex fonts is not in place, and the font deformation seriously loses the original font structure.

In addition to improved encoder functionality using the introduction of deformable convolution, we find that surface subdivision artifacts appear in the input residual module. This feature can carry the edge features at the edges of the font that are not recognized by the network, which can lead to a style overflow in the network after style migration. To address this issue, we increase the depth of the residual module to further control the font deformation strength.

### 3.3. Transfer Networks ( $G_T$ )

For the  $G_T$  module, we use the texture network structure of the shape-matching GAN [1] model. After the glyph generation network, we obtain a text mask style image with the learned style features. Similar to the glyph generation network  $G_S$ , a large number of data pairs are obtained by clipping the style image and the text mask image, and a large number of data pairs are trained to quickly build an end-to-end fast text style model so that the style network can adapt to the shape of text and quickly generate target images. The network can generate multiple styles of text images and easily control the style of the text. The main idea is that by taking the deformed text images with style characteristics that we have generated as input for the transmission network, we can select the style images that need to be migrated, and all text images can be effectively trained in the network to obtain the corresponding style images. The advantage of this network is that multiple text styles can be generated using a single model, and the generation of text styles can be controlled.

### 3.4. Loss Function

The loss of the network  $G_S$  contains a reconstruction loss and an adversarial loss. In the reconstruction loss,  $l(l \in [0, 1])$  controls the degree of deformation. Set  $l$  to control font deformation and to realize multiscale style migration.  $x$  represents the structural sketch obtained after a binary transformation of the style image, and  $y$  represents a raw style image.  $\tilde{x}_i$  represents the result of style structure images with different degrees of deformation obtained from the  $G_B$  network. We use a mask image as an information guide to reconstruct the structure of the different style images. The reconstruction loss restores the structure of the different degrees of images for each style to the structure of the original. In the adversarial loss, we add the mask images to the generator and the discriminator, similar to the conditional GAN procedure.

$$\mathcal{L}_s^{rec} = \sum_{i=1}^N \mathbb{E}_{x,l,mask} \left[ \left\| G_S(\tilde{x}_i, l, mask_i) - x_i \right\|_1 \right] \quad (5)$$

$$\mathcal{L}_s^{adv} = \sum_{i=1}^N \mathbb{E}_{x,mask} [\log D_S(x_i, mask_i)] + \sum_{i=1}^N \mathbb{E}_{x,l,mask} \left[ \log \left( 1 - D_S \left( G_S(\tilde{x}_i, l, mask_i) \right) \right) \right] \quad (6)$$

The overall  $G_S$  loss is as follows:

$$\mathcal{L}_{G_S} = \min_{G_S} \max_{D_S} \lambda_s^{adv} \mathcal{L}_s^{adv} + \lambda_s^{rec} \mathcal{L}_s^{rec} \quad (7)$$

The main task of the  $G_T$  network is to assign texture features to the structural images obtained in  $G_S$ . The loss of the network  $G_T$  includes reconstruction loss, conditional

adversarial loss, style loss, and texture loss. Style loss  $\mathcal{L}_T^{sty}$  is proposed in the neural style transfer.

$$\mathcal{L}_T^{rec} = \sum_{i=1}^N \mathbb{E}_{x,y,mask} [\| G_T(x_i, mask_i) - y_i \|_1] \tag{8}$$

$$\mathcal{L}_T^{adv} = \sum_{i=1}^N \mathbb{E}_{x,mask,y} [\log D_T(x_i, mask_i, y_i)] + \sum_{i=1}^N \mathbb{E}_{x,l,mask} [\log(1 - D_T(G_T(x_i, mask_i)))] \tag{9}$$

The overall  $G_T$  loss is as follows:

$$\mathcal{L}_{G_T} = \min_{G_T} \max_{D_T} \lambda_T^{adv} \mathcal{L}_T^{adv} + \lambda_T^{rec} \mathcal{L}_T^{rec} + \lambda_T^{sty} \mathcal{L}_T^{sty} + \lambda_T^{tex} \mathcal{L}_T^{tex} \tag{10}$$

The loss function described above applies to our basic networks,  $G_S$  and  $G_T$ . In the sketch model, we first select a text image  $t$  as the base image and randomly select an  $l$  value within  $[0, 1]$  to reconstruct image  $t$ .

$$\mathcal{L}_B^{rec} = \mathbb{E}_{t,l} [\| G_B(t, l) - t \|_1] \tag{11}$$

After obtaining the reconstructed image  $t$ , we generate an adversarial loss function to make the reconstructed image more similar to the original image.

$$\mathcal{L}_B^{adv} = \mathbb{E}_{t,l} \left[ \log D_B \left( t, l, \bar{t}_l \right) \right] + \mathbb{E}_{t,l} \left[ \log \left( 1 - D_B \left( G_B(t, l), l, \bar{t}_l \right) \right) \right] \tag{12}$$

When the target image is smoothed by the sketch model for edge features, the image is not smoothed well due to the influence of the recovery algorithm on the noise amplification, which causes some features to be lost and additional noise features to be added to our image when it is input in  $G_S$ . Consequently, in the process of migrating the resultant image of  $G_S$  for stylistic features, a small amount of noise has a great impact on the result, resulting in shadows at the edges of the images, and the total proportion of images contaminated by noise is significantly larger than the proportion of noise-free images. Therefore, we design a new smooth loss by adding regular terms in the sketch model to maintain the smoothness of the image. The difference in adjacent pixel values in the image can be solved to some extent by reducing the loss, and our loss solves the edge shading problem. We also implement the noise constraint by sacrificing image sharpness, which finally solves the problem of noise and poor edge smoothing in the image. The following equation is the regular term that we add.

$$\mathfrak{R}_V \beta(x) = \sum_{i,j} \left( (x_{i,j-1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\frac{\beta}{2}} \tag{13}$$

The overall sketch model loss is as follows:

$$\mathcal{L}_{Smooth} = \min_{G_B} \max_{D_B} \lambda_B^{adv} \mathcal{L}_B^{adv} + \lambda_B^{rec} \mathcal{L}_B^{rec} + \mathfrak{R}_V \beta(x) \tag{14}$$

## 4. Experiment

### 4.1. Dataset

We use the dataset TE141K [37] which contains 152 professionally designed text effects rendered on glyphs, including English letters, Chinese characters, and Arabic numerals. The dataset is divided according to the 8:2 ratio, including 608 pictures in the training set and 152 pictures in the test set. This dataset is one of the largest font style migration datasets to date and can be used in research areas such as font style migration, multidomain transfer, and image-to-image translation.

#### 4.2. Training

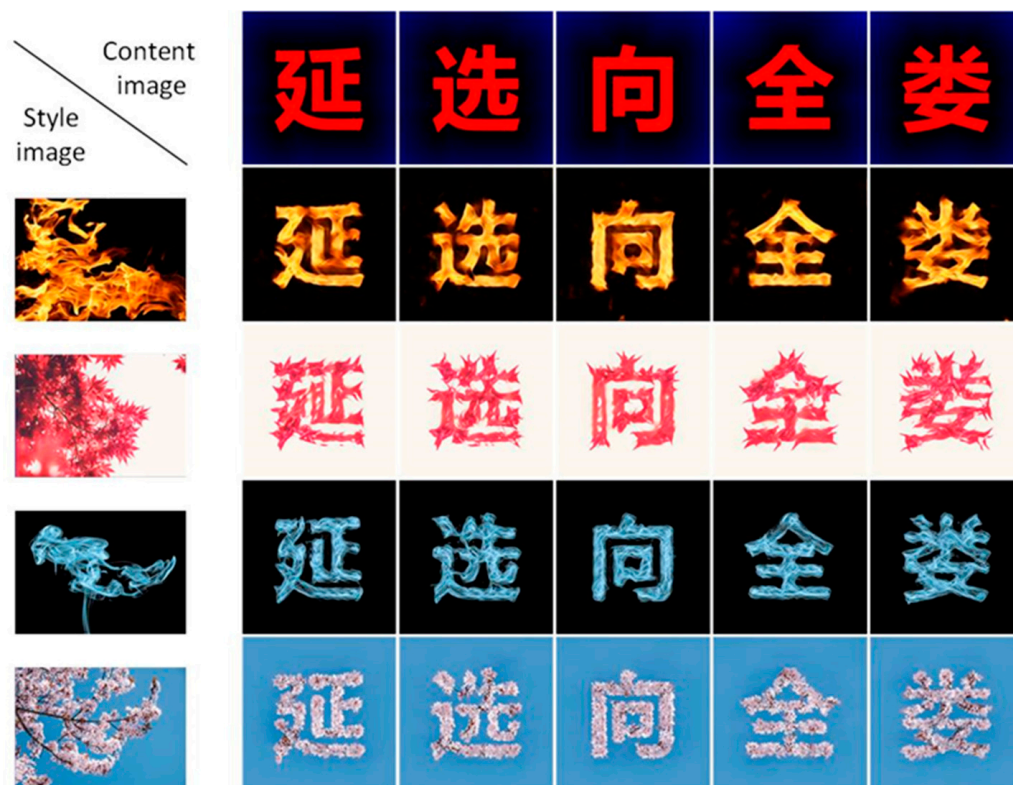
Our model consists of the sketch module  $G_B$ , glyph module  $G_S$  and transfer module  $G_T$ , so we divide the training strategy into three steps and randomly crop the images to a  $256 \times 256$  image size before the training starts. For the optimizer, we use the Adam optimizer and set the learning rate to 0.0002. We perform 3 training epochs. First, we need only input a style image mask to train the sketch module  $G_B$ . Then, the model smooths the input image to reduce the sharpening of the image edges, and in this process, the smoothing effect of the network on the image is further improved by the smooth loss we design. We need the model to connect the source style domain and the target text domain using a smoothing block, which maps the style image and the style image mask to the smoothing domain, where the details are eliminated, and the contours show a similar degree of smoothing. According to the adjustment parameter  $l$  ( $l \in [0, 1]$ ), the smoothed style image is transformed into different degrees of a mask. Next, we train the  $G_S$ . By clipping the mask, the training pairs of sketch shapes with different smoothness can be obtained. The training pair is fed into the  $G_S$  network, and the glyph network  $G_S$  is trained to map it to the original text mask so that  $G_S$  can characterize the shape features of text image mask and transfer these features to the target text. The encoder we design can more flexibly control font deformation at different levels and enhance the model generalization ability. The dilated convolution, deformable convolution and residual block structure we design make the edges of stylized images more convergent to the edges of text images and font deformation more flexible and controllable. Finally, we train the  $G_T$  module. Here, it is necessary to randomly cut a style image mask and a style image to form a training pair as the input for  $G_T$ . The network  $G_T$  is trained to perform texture rendering instead. Style migration on the input image is performed so that the deformed font has the style features of the style image.

#### 4.3. Comparisons with State-of-the-Art Methods

We used shape-matching GAN as the baseline and conducted a number of experiments. The effects of our proposed method on artistic text style transfer are shown in Figure 4. On the one hand, our method is superior to the baseline at stylizing complex glyphs. On the other hand, our method represents a significant improvement over the baseline method for complex glyphs, ensuring a clear font structure and improving legibility.

**Effect picture comparison.** In Figure 5, we qualitatively compare our method with four state-of-the-art style transfer methods, neural style transfer (NST) [21], LapStyle [38], multi-style transfer (MST) [36], and shape-matching GAN [3]. These methods are chosen because they are all one-way style transfers, and most style transfer methods are derivative versions of these methods. (a) NST is the most basic style transfer, which uses a CNN for feature extraction and then uses the extracted features for reconstruction. It can transfer the style but cannot learn the style features, and the glyphs are homogenized. (b) LapStyle splits the complex style migration into an initial migration at low resolution and a correction process at high resolution, which effectively improves the quality and the speed of stylization. Thus, LapStyle transfer is more suitable for overall image style migration. However, this method is not applicable to artistic font text generation because it is ineffective in extracting the features of fonts, which represent only one aspect of text images. (c) MST is a recently proposed and diversified transformation network for text style transfer that can generate multiple text images in a single model and control the text style in a simple way. (d) Shape-matching GAN is our baseline method, which cannot maintain the structure of the complex font glyphs. As seen from the results of the comparative experiments in Figure 5, our proposed method has obvious advantages in terms of the effectiveness of artistic font generation. Most other methods involve style transfer of the whole style image and thus have an insufficient feature extraction effect on text and style images, which leads to the inability to generate clear and beautiful artistic text. In contrast, by introducing a deformable convolution and an improved residual module, our proposed network enhances the control of font deformation, enabling a more detailed font

feature extraction effect and solving the problems of severe font deformation and unclear character shapes. It differs from other style transfer networks in that the text has texture details while learning the image style, making the generated artistic characters nondual and artistically ornamental.



**Figure 4.** Our artistic text style transfer effects.

**Execution time comparison.** We compare the time needed to generate an image of different models in the testing process with Intel Core i7-11700k 3080 10G, as shown in Table 1. We input  $320 \times 320$  images into the model and average the reasoning time required for 100 pictures. As seen from Table 1, each image generated by our proposed mode requires only 0.039 s on average, and we can nearly interact with users in real time. Our time is slightly longer than that of shape-matching GAN [3] because of the addition of deformable convolution to the model. Deformable convolution adds only a small overhead for the model parameters and computation. However, it is precisely because of deformable convolution that our model can better capture the edge features of fonts and produce better results. NST [22] takes a long time to execute because it requires several iterations during testing to generate the final result.

**Table 1.** Execution time comparison.

Model	Execution Time(s)
NST [19]	62
LapStyle [35]	5
MST [33]	6
Shape-Matching GAN [1]	0.034
Ours	0.039





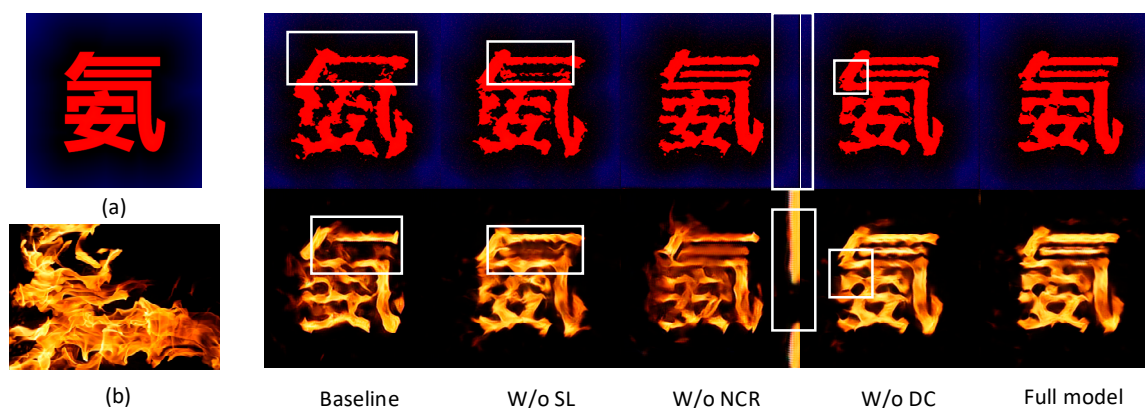
Figure 5. Comparison with state-of-the-art methods on various styles.

#### 4.4. Ablation Study

To analyze the advantages of our improvements on the baseline model, we design the following experiments with different configurations:

- Baseline: Our baseline network uses the original shape-matching GAN approach [3] trained to directly map the structure map  $X$  back to the style image  $Y$ .
- W/o SL: The model adds a smooth loss (SL) to improve the smoothing performance of the sketch module.
- W/o NCR: The model adds a new controllable ResBlock (NCR) to improve the baseline shape-matching GAN model.
- W/o DC: This model only adds a deformable convolution (DC) to the encoder without NCR.
- Full model: The proposed model incorporates our redesigned encoder (DR), which includes an NCR and DC.

The results of this ablation experiment are shown in Figure 6. It can be seen that compared with the baseline network, W/o SL enhances the smoothing performance, which can make the text better learn style features and maintain the font. The W/o NCR model improves the legibility of the font and can guarantee the structural features of the font. However, the edge features are recognized, resulting in style overflow. Therefore, due to the flexibility of deformable convolution in feature extraction, the W/o DC model solves the problem of style overflow caused by identifying unnecessary edge features. In sum, when we adopt the full model, the results effectively solve the problem of the missing glyph structure and greatly increase the visibility of the text.



**Figure 6.** Comparison chart of ablation experiments: (a) represents our original data, and (b) is the style features we need to migrate. The first row on the right is the resulting graph of the texture generation network  $G_S$ , and the second row is the final output graph. From left to right are the output results of the model we proposed above.

#### 5. Conclusions

In this paper, we propose the deep deformable artistic font style transfer network that maps the stylistic features of an image to the text of a text image and controls the degree of font deformation by adjusting parameters to achieve diverse style migration. In the network, the DR encoder that we designed can effectively extract font features, control font deformation, greatly improve the recognition accuracy of complex fonts, and enable the network to generate more exquisite art fonts. The DAF network is divided into three modules, and each module can be trained separately. In the sketch module, smooth loss is introduced to enhance the smoothness of the font edges and improve the similarity between the font edges and the edge transformations of the style images. In the  $G_S$  module, the novel DR encoder is used to better preserve the font structure and improve font legibility. The  $G_T$  module is trained to transfer the style image features to the font image so that the font not only retains its own glyph structure but also integrates the style features. We

verified the effectiveness and robustness of the method by comparing it with state-of-the-art migration algorithms. In future work, we hope to integrate the attention mechanism with the DR encoder to improve font adaptivity, which will make the font style transfer more precise for text, resulting in more beautifully migrated text. Additionally, we will work on research measuring an improvement in contour definition.

**Author Contributions:** Conceptualization, X.Z. and M.L.; methodology, X.Z. and M.L.; software, X.Z. and K.W.; validation, X.Z. and K.W.; formal analysis, M.L.; writing, X.Z.; review and editing, M.L., H.Z. and X.S.; visualization, X.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by the Scientific Research Fund of Hunan Provincial Education Department (22A0502), the National Natural Science Foundation of China (61772179), the Hunan Provincial Natural Science Foundation of China (2019JJ40005), the 14th Five-Year Plan Key Disciplines and Application-oriented Special Disciplines of Hunan Province (Xiangjiaotong [2022] 351), the Science and Technology Plan Project of Hunan Province (2016TP1020), the Science and Technology Innovation Project of Hengyang(202250045231), the Open Fund Project of Hunan Provincial Key Laboratory of Intelligent Information Processing and Application for Hengyang Normal University (2022HSKFJ012), and the Postgraduate Scientific Research Innovation Project of Hunan Province (QL20210262).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, K.; Yu, J. The Computer-Based Generation of Fonts in the Style of Kandinsk. *Leonardo* **2021**, *5*, 437–443. [[CrossRef](#)]
2. Gupta, R.; Srivastava, D.; Sahu, M.; Tiwari, S.; Ambasta, R.K.; Kumar, P. Artificial intelligence to deep learning: Machine intelligence approach for drug discovery. *Mol. Divers.* **2021**, *25*, 1315–1360. [[CrossRef](#)] [[PubMed](#)]
3. Yang, S.; Wang, Z.; Wang, Z.; Xu, N.; Liu, J.; Guo, Z. Controllable artistic text style transfer via shape-matching gan. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4442–4451.
4. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017; pp. 764–773.
5. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019), Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
6. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
7. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)] [[PubMed](#)]
8. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **1980**, *36*, 193–202. [[CrossRef](#)] [[PubMed](#)]
9. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
10. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
11. Shinde, P.P.; Shah, S. A review of machine learning and deep learning applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA 2018), Pune, India, 16–18 August 2018; pp. 1–6.
12. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [[CrossRef](#)]
13. Liu, Z.; Lin, W.; Li, X.; Rao, Q.; Jiang, T.; Han, M.; Fan, H.; Sun, J.; Liu, S. ADNet: Attention-guided deformable convolutional network for high dynamic range imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021), virtual, 19–25 June 2021; pp. 463–470.
14. Zhao, C.; Zhu, W.; Feng, S. Superpixel guided deformable convolution network for hyperspectral image classification. *IEEE Trans. Image Process.* **2022**, *31*, 3838–3851. [[CrossRef](#)] [[PubMed](#)]
15. Luo, Z.; Li, Y.; Cheng, S.; Yu, L.; Wu, Q.; Wen, Z.; Fan, H.; Sun, J.; Liu, S. BSRT: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022), New Orleans, LA, USA, 21–24 June 2022; pp. 998–1008.
16. Shi, Z.; Liu, X.; Shi, K.; Dai, L.; Chen, J. Video frame interpolation via generalized deformable convolution. *IEEE Trans. Multimed.* **2021**, *24*, 426–439. [[CrossRef](#)]



17. Chen, J.; Pan, Y.; Li, Y.; Yao, T.; Chao, H.; Mei, T. Retrieval Augmented Convolutional Encoder-Decoder Networks for Video Captioning. *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**, *19*, 1–24. [[CrossRef](#)]
18. Chen, F.; Wu, F.; Xu, J.; Gao, G.; Ge, Q.; Jing, X.Y. Adaptive deformable convolutional network. *Neurocomputing* **2021**, *453*, 853–864. [[CrossRef](#)]
19. Xu, S.; Zhang, L.; Huang, W.; Wu, H.; Song, A. Deformable convolutional networks for multimodal human activity recognition using wearable sensors. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 2505414. [[CrossRef](#)]
20. Park, J.; Yoo, S.; Park, J.; Kim, H.J. Deformable graph convolutional networks. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2022), virtual, 22 February–1 March 2022; pp. 7949–7956.
21. Yu, B.; Jiao, L.; Liu, X.; Li, L.; Liu, F.; Yang, S.; Tang, X. Entire Deformable ConvNets for semantic segmentation. *Knowl.-Based Syst.* **2022**, *250*, 108871. [[CrossRef](#)]
22. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2414–2423.
23. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference (ECCV 2016), Amsterdam, The Netherlands, 11–14 October 2016; pp. 694–711.
24. Luan, F.; Paris, S.; Shechtman, E.; Bala, K. Deep photo style transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 22–25 July 2017; pp. 4990–4998.
25. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Proc. Mag.* **2018**, *35*, 53–65. [[CrossRef](#)]
26. Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Yu, Y.; Song, M. Neural style transfer: A review. *IEEE Trans. Vis. Comput. Graph.* **2019**, *26*, 3365–3385. [[CrossRef](#)] [[PubMed](#)]
27. Singh, A.; Jaiswal, V.; Joshi, G.; Sanjeeve, A.; Gite, S.; Kotecha, K. Neural style transfer: A critical review. *IEEE Access* **2021**, *9*, 131583–131613. [[CrossRef](#)]
28. Liu, M.Y.; Tuzel, O. Coupled generative adversarial networks. In Proceedings of the Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 1–29.
29. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017; pp. 2223–2232.
30. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International conference on machine learning (ICML 2017), Sydney, NSW, Australia, 6–11 August 2017; pp. 214–223.
31. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2–4 May 2016; pp. 1–19.
32. Liu, X.; Meng, G.; Chang, J.; Hu, R.; Xiang, S.; Pan, C. Decoupled representation learning for character glyph synthesis. *IEEE Trans. Multimedia* **2021**, *24*, 1787–1799. [[CrossRef](#)]
33. Azadi, S.; Fisher, M.; Kim, V.G.; Wang, Z.; Shechtman, E.; Darrell, T. Multi-content gan for few-shot font style transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7564–7573.
34. Yang, S.; Wang, Z.; Liu, J. Shape-Matching GAN++: Scale controllable dynamic artistic text style transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3807–3820. [[CrossRef](#)] [[PubMed](#)]
35. Zhang, H.; Dana, K. Multi-style generative network for real-time transfer. In Proceedings of the European Conference on Computer Vision (ECCV 2018) Workshops, Munich, Germany, 8–14 September 2018; pp. 349–365.
36. Yuan, H.; Yanai, K. Multi-style transfer generative adversarial network for text images. In Proceedings of the 4th IEEE International Conference on Multimedia Information Processing and Retrieval (MIPR 2021), Tokyo, Japan, 8–10 September 2021; pp. 63–69.
37. Yang, S.; Wang, W.; Liu, J. Te141k: Artistic text benchmark for text effect transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3709–3723. [[CrossRef](#)] [[PubMed](#)]
38. Lin, T.; Ma, Z.; Li, F.; He, D.; Li, X.; Ding, E.; Wang, N.; Li, J.; Gao, X. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021), virtual, 19–25 June 2021; pp. 5141–5150.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.