

RESEARCH ARTICLE

An improved defocusing adaptive style transfer method based on a stroke pyramid

Jianfang Cao^{1,2*}, Zeyu Chen^{1,2}, Mengyan Jin^{1,2}, Yun Tian¹

1 Department of Computer Science & Technology, Xinzhou Normal University, Xinzhou, China, **2** School of Computer Science & Technology, Taiyuan University of Science and Technology, Taiyuan, China

* caojianfangcn@163.com

Abstract

Image style transfer aims to assign a specified artist's style to a real image. However, most existing methods cannot generate textures of various thicknesses due to the rich semantic information of the input image. The image loses some semantic information through style transfer with a uniform stroke size. To address the above problems, we propose an improved multi-stroke defocus adaptive style transfer framework based on a stroke pyramid, which mainly fuses various stroke sizes in the image spatial dimension to enhance the image content interpretability. We expand the receptive field of each branch and then fuse the features generated by the multiple branches based on defocus degree. Finally, we add an additional loss term to enhance the structural features of the generated image. The proposed model is trained using the Common Objects in Context (COCO) and Synthetic Depth of Field (SYNDOF) datasets, and the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are used to evaluate the overall quality of the output image and its structural similarity with the content image, respectively. To validate the feasibility of the proposed algorithm, we compare the average PSNR and SSIM values of the output of the modified model and those of the original model. The experimental results show that the modified model improves the PSNR and SSIM values of the outputs by 1.43 and 0.12 on average, respectively. Compared with the single-stroke style transfer method, the framework proposed in this study improves the readability of the output images with more abundant visual expression.



OPEN ACCESS

Citation: Cao J, Chen Z, Jin M, Tian Y (2023) An improved defocusing adaptive style transfer method based on a stroke pyramid. PLoS ONE 18(4): e0284742. <https://doi.org/10.1371/journal.pone.0284742>

Editor: Mohamed Hammad, Menoufia University, EGYPT

Received: January 31, 2023

Accepted: April 6, 2023

Published: April 24, 2023

Copyright: © 2023 Cao et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper.

Funding: This study was funded by the Humanities and Social Sciences Research Project of the Ministry of Education (Planning Fund Project) (21YJAZH002).

Competing interests: The authors have declared that no competing interests exist.

1. Introduction

Artificial intelligence (AI) has attracted considerable attention due to its powerful capacity in multiple application fields, such as intelligent processing and self-adaptive learning. AI can handle vast quantities of data and solve complex questions through algorithm and model training and optimization. AI is also equipped with intelligent self-learning, which enables it to gradually improve its self-learning and judgment capacities based on data and experience accumulation and analysis. These virtues guarantee its cross integration in multiple fields. With the rapid development of AI technology represented by deep learning, an increasing number of fields have begun to cross integrate, such as groundwater storage modeling [1],

climate change forecasting [2], environmental factor analysis [3], air pollution prediction [4], satellite image classification [5], weed detection [6], forest area classification [7], agricultural water resource management [8], biomedical fields [9–11] and biometric identification [12]. These studies have brought great convenience and improvement to people's lives and work. Therefore, AI has attracted considerable attention in the current scientific and technological fields and has great development prospects.

Style transfer is also a typical practice of integrating art and deep learning technology. Specifically, given a content graph and a target style graph, the purpose of style transfer is to transform the style of the original content graph into the style of the target graph while ensuring that the semantic information of the original content is not lost. At present, all kinds of drawing software, represented by mobile phone image applications, use a large number of related technologies to provide people with leisure and entertainment functions. Inspired by convolutional neural networks (CNNs), Gatys et al. [13] first studied how CNNs can be used to recreate pictures in natural famous painting styles. They suggested modeling photo content as feature responses from a pretrained CNN, and others have modeled art styles as summary feature statistics. Subsequently, the model-based iterative optimization method proposed by scholars improved the image stylization efficiency. Li et al. [14] proposed replacing the Gram matrix matching in the Gatys model with the Markov regularization model by combining the Markov random field and a VGG network. Johnson et al. [15] and Ulyanov et al. [16] proposed a fast neural style transfer model that achieves real-time stylization by training forward neural networks. Although the above generation model method is two orders of magnitude faster than the previous style transfer method based on image iteration, it can generate images of only specific styles. To obtain other styles, a feedforward generation network needs to be retrained, so it is less flexible and very time consuming. Therefore, the single model multistyle generation network began to appear, which integrates multiple styles into one model to improve the efficiency of the feedforward network. Zhang et al. [17] proposed the concept of the CoMatch layer, which requires the model to learn multiple styles and then use the target style image features as a signal input to guide the input image to match the style features in the CoMatch layer to achieve style transfer. He et al. [18] proposed the deep-exemplar-based method. By using a large sample database in combination with the input reference image, a style transfer scheme is generated after comprehensive calculation. Li et al. [19] proposed a style selection model containing multiple styles, which uses image pixels as a signal input to control stylized image generation. This model can synthesize more than 300 textures and can generate 16 styles of feedforward networks. The above method is more flexible, but a single network can achieve only limited categories of style conversion. Chen et al. [20] proposed using a method called "style swap" to achieve any style transfer model. The method uses a pretrained VGG network to extract content and style features from the input information, divides them into activation blocks of the same size, and then matches each content activation block with the most similar style activation block to obtain a generated image. The model has high flexibility, but the stylization speed is slow. Huang et al. [21] proposed a method called the adaptive instance normalization (AdaIn) layer. The algorithm calculates the mean and variance in the extracted content features and style features according to the channel dimension, applies the mean and variance in the style features to the content features, and reconstructs the resulting image through the decoder. Li et al. [22] introduced the whiten-color transformation (WCT) mechanism into the style transfer algorithm. This method stripped the original color information of the content features and then used the color information of the style features to achieve the transfer effect. Park et al. [23] proposed a self-attention network (SANet), which uses a self-attention mechanism to mix content features and style features and then flexibly modifies local styles according to the semantic spatial distribution of the content images. In

addition, some methods add a 'stroke control' function to the original style transfer algorithm. A single model can generate not only good quality style transfer results but also control the thickness of textures in the image, thus providing users with more choices. Jing et al. [24] proposed a network structure called the stroke pyramid. By limiting the correspondence between the stroke size and the style image size, the algorithm uses convolutional layers with different receptive fields as different stroke branches to achieve continuous stroke size control in the spatial dimension. Yao et al. [25] proposed an arbitrary style transfer architecture with multi-stroke fusion. The algorithm uses multi-scale style swaps to generate multiple stroke features and then fuses each stroke feature through self-attention maps to achieve adaptive coordination of style and content in the spatial dimension. Recently, an increasing number of reports on the cross-fusion between style transfer algorithms and classic models have emerged. Zhang et al. [26] proposed a hierarchical vision transformer using strip window attention. This approach realizes accurate style transfer by focusing on local image domains and adapting to a wide range of styles. Feng et al. [27] utilized a synthesized transformer-based automatic encoder for style transfer. This approach combines an optimization strategy to learn the style and content representations contained in images to exert accurate control over the style transmission process.

However, most existing algorithms focus only on generating images with similar styles, ignoring the destruction of image semantics in the style transfer process. For the feedforward network model trained for style images with many lines and dense textures, the saliency of the texture-expressing style in the generated image is higher than that of the texture-expressing content, thus masking the content information to be expressed by the image, making it impossible or difficult for readers to interpret the content in the image. Inspired by the gold tower of strokes, we use strokes of various sizes to depict the generated images to enhance the semantic saliency of the images. We use the defocus estimation algorithm to predict the clarity of the image in pixels to obtain the distribution of clear areas. The clear region is usually the content focus of the image, which contains rich semantic information. In contrast, the fuzzy region expresses less semantic information. We use smaller strokes to depict the clear areas, that is, to retain as many detailed features as possible contained in the original image, and use larger strokes to depict blurred areas to distinguish them from clear areas, increasing the saliency of the main image content. Additionally, we improve the network structure of the stroke gold tower. The receptive field of each branch of the original stroke pyramid is small, and it cannot accommodate large geometric microstructures during the training process, resulting in a reduced style effect. Therefore, we use dilated convolution to make each branch obtain a larger receptive field. To address the problem of jagged traces caused by the superposition of multiple dilated convolutions, we use various sizes of dilated convolution kernels. The defocus blur estimation algorithm is a method used to predict the blur degree of each pixel in a real image. The traditional method is to generate a complete defocus map by iteration. Zhuo et al. [28] and Karaali et al. [29] calculated the ratio between the gradient of the input image and the re-blurred image as the blurring amount of the edge position; then, they propagated the blurring amount of the edge to the adjacent pixels by an iterative method to calculate the complete defocus image. Shi et al. [30] used a sparse fuzzy scale instead of a fuzzy quantity to represent the blur degree of pixels. Tang et al. [31] used spectral amplitude as a clue to estimate the blur amount and estimated the blur amount at these edge locations by establishing the relationship between the amount of spatially varying defocus blur and spectrum contrast at edge locations. Xu et al. [32] divided the image into local image blocks to calculate the blur amount and to improve prediction efficiency. Park et al. [33] proposed a multi-scale image block extraction strategy to reduce the algorithm dependence on the image block size, thereby improving the synthesis result quality. However, the iterative algorithm is time consuming, and the defocus

estimation algorithm based on deep learning was proposed. Lee et al. [34] integrated the Synthetic Depth-of-Field (SYNDOF) dataset, constructed a decoder corresponding to VGG-Net and introduced the domain adaptation method to improve the defocus estimation efficiency. We fuse the feature results output by the multiple branches and the defocused image according to the clarity of the spatial dimension so that the algorithm can weigh the size of the stroke according to the richness of the semantic information expressed by the image.

In this paper, we propose an improved multi-stroke defocus adaptive style transfer method based on a stroke pyramid. The main contributions include the following aspects.

1. A depth estimation based on defocus blur is introduced, and the feature maps of various stroke sizes are combined by using the feature weights after soft threshold segmentation. The framework of this paper can adaptively adjust the strokes and textures of the output image according to the blur degree of the image. The difference between the defocused blurred area and the focused clear area after stylization is more obvious. The main body of the image can be better expressed and has richer textures.
2. By using residual blocks composed of dilated convolutions to expand the receptive field of each stroke pyramid branch, the network can learn to depict larger strokes.
3. Gram loss and mean standard deviation loss are used as the evaluation criteria for style similarity so that the generated image can better retain the structural information from the style image.

This article is divided into three parts. In the following section, we introduce the theoretical basis of the algorithm, network structure, loss function calculation and the experiment performed in this study. In the third section, we focus on the results obtained in this study, and comparisons are also made between our model and those reported in the literature. In the last section, we summarize and discuss the results of this study and propose future research directions for style transfer algorithms.

2. Methodology

2.1 Background theory

2.1.1 MobileNetV3 feature extraction network. The MobileNetV3 model is a 20-layer convolutional neural network for extracting features, and part of its structure is shown in Table 1. The network adds a squeeze-and-excitation (SE) network to the core architecture and improves the quality of the features extracted by the network by showing the interdependence between the convolutional feature channels in its modeling network.

The SE mechanism automatically acquires the importance of each feature channel by learning and then, based on those results, enhances the useful features while it suppresses features

Table 1. MobileNetV3 feature extraction network structure.

Input	Operator	outChannel	stride
224×224×3	conv2d	16	2
112×112×16	bneck_1_1, 3×3	16	1
112×112×16	bneck_1_2, 3×3	24	2
56×56×24	bneck_2_1, 3×3	24	1
56×56×24	bneck_2_2, 5×5	40	2
28×28×40	bneck_3_1, 5×5	40	1
28×28×40	bneck_3_2, 5×5	40	1
28×28×40	bneck_3_3, 3×3	80	2

<https://doi.org/10.1371/journal.pone.0284742.t001>

that are not useful for the current task; the network can learn to selectively emphasize or suppress certain feature information by using global information. Due to the need for different processing of image feature information, this paper uses the pretrained MobileNetV3 to realize the feature extraction operation of the generated network. Compared with the VGG feature extraction network, MobileNetV3 has fewer convolution layers and network parameters, making it more lightweight, thus reducing the calculations required of the network. In this paper, we use the output of the `bneck_2_1` layer as image feature information. Compared with the original image size, this feature is twice as small in size, containing only 24 channels. The smaller number of channels ensures the weight of the subsequent network. Different subsequent processing branches can expand the channels as required to improve the network performance.

2.1.2 Stroke pyramid. The stroke pyramid was first proposed by Jing et al. [24], who corresponded style images of different sizes to strokes of different sizes for network training. Although two images generated with two different strokes show the same semantics and style, they have extensive and delicate differences in texture structure. In the style transfer algorithm, the style similarity between the style image and the output image is calculated by the Gram matrix. The Gram matrix is sensitive to the size of strokes, and therefore, a change in the size of the style image is equivalent to a change in the stroke size of the image; thus, the trained generative network generates different size strokes.

By controlling the dependence between the receptive field of the generative network and the size of the style image, continuous multi-stroke style transfer in the spatial dimension is realized. As shown in Fig 1, the stroke pyramid network is divided into a decoder encoder, encoder decoder, stroke pyramid migration network, and VGG-Net loss network. The decoder consists of three convolutions, which are responsible for downsampling the input image to extract image features. The encoder is composed of one convolution and two deconvolutions and is responsible for reconstructing image features into output images. The shallow network has a smaller receptive field and is better at learning to draw small strokes. In contrast, a deep network with a larger receptive field is better at learning to draw large strokes. Since the output

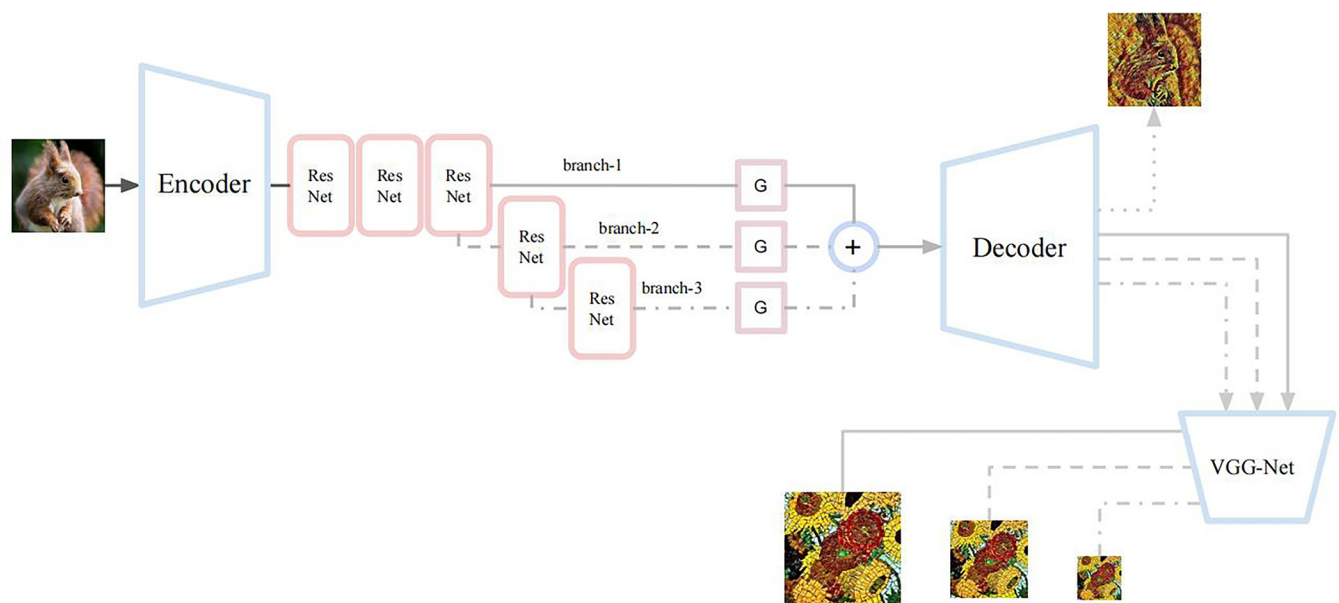


Fig 1. Stroke pyramid network structure.

<https://doi.org/10.1371/journal.pone.0284742.g001>

of each branch is the input of the previous branch, the output of the different branches has a certain correlation in space, which is conducive to the expression of multiple strokes in space.

2.2 Method description

The workflow of the improved defocus adaptive style migration method based on the stroke pyramid is shown in Fig 2. First, we preprocess the image that needs to be stylized, adjust the mean and standard deviation of the image in the channel dimension, and then extract the high-dimensional image features through an encoder and perform style migration and defocus estimation processing for the high-dimensional features. Finally, we fuse the stylized features based on defocus information and obtain the stylized image through decoding operations.

2.2.1 Network overall design. The defocusing adaptive multi-stroke network is shown in Fig 3. The design criterion of the model is to design a larger receptive field on each branch of the stroke pyramid so that the network can learn larger strokes while ensuring that the network layer is not too deep; a shallower network layer ensures the continuity of adjacent strokes in space. At the same time, defocus estimation is used to detect the distribution of clear and blurred areas in the image, and multiple strokes are fused with this as a reference to highlight the main area in the output image and to create a better migration effect.

The model structure shown in Fig 3 is comprises an encoder, a decoder, a style converter, a defocus blur estimation module, a soft mask gating module and a loss network. The encoder comprises the first four layers of MobileNet v3 and is responsible for encoding the image into a 24-channel high-dimensional feature map. Then, the high-dimensional feature maps are passed to the style converter and the defocus estimation module. The style converter can make the current feature obtain a specific style, which comprises several residual blocks with dilated convolution, and the number of channels in each residual block is 48. When considering the problem of generating the receptive field of the network, the first two residual blocks are used to improve the depth of the network, and the remaining residual blocks form independent branches. The features output by each branch skip the subsequent residual blocks and directly enter the decoder. Such structural division enables the different branches to extract feature maps under various receptive fields. The defocus blur estimation module converts the received image features into a single-channel defocus map that is 4 times smaller than the original image size. Each code value of the defocus map belongs to the interval $[0,1]$, which represents the degree of blurring at this position. Then, the features of multiple branch outputs are fused

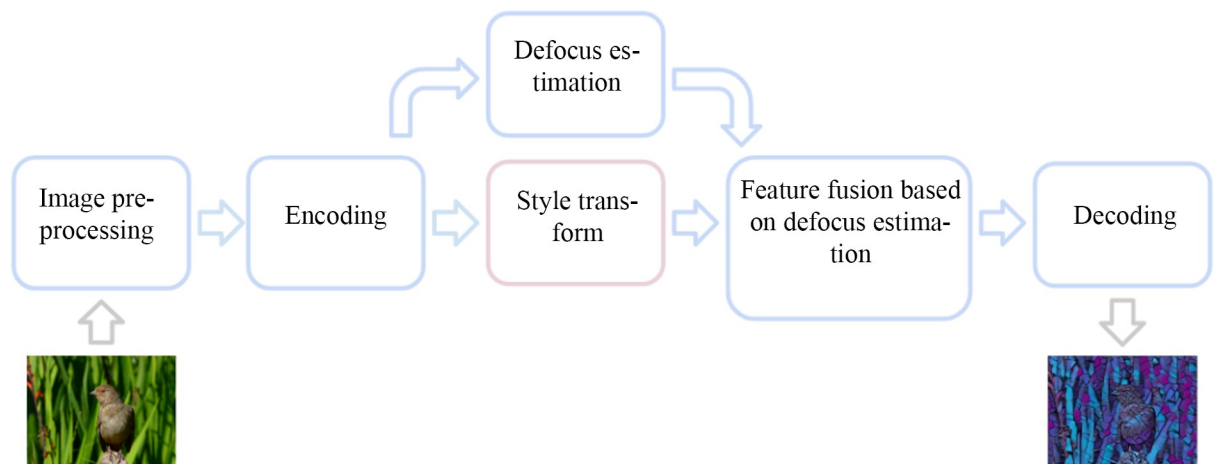


Fig 2. Workflow of the improved defocus adaptive style migration method based on the stroke pyramid.

<https://doi.org/10.1371/journal.pone.0284742.g002>

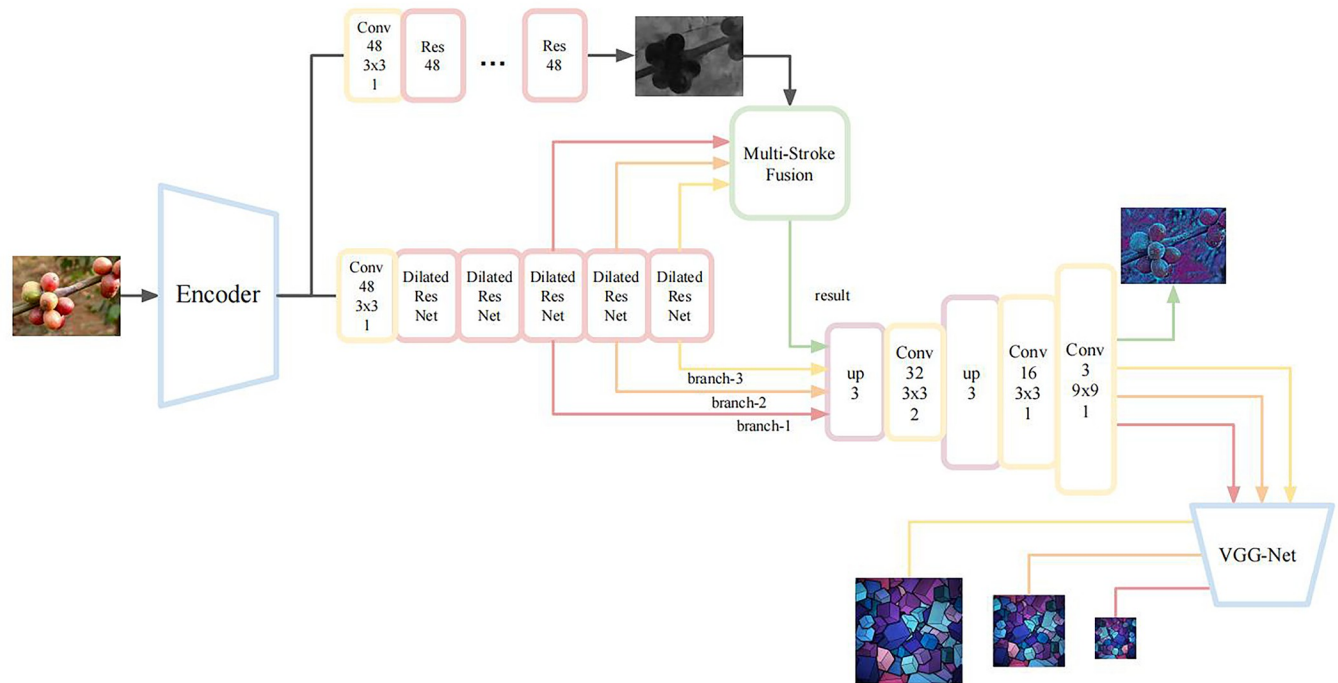


Fig 3. Defocusing adaptive multi-stroke network structure.

<https://doi.org/10.1371/journal.pone.0284742.g003>

with reference to the defocus map. The decoder consists of two upsampling layers and a convolution layer, which is responsible for reconstructing the feature map into a stylized image. Considering the checkerboard effect of deconvolution, nearest neighbor interpolation is used instead of deconvolution as the upsampling layer. When training the network, the output image, the style image, and the content image are transmitted to the VGG-19 network, and the extracted layer features are used to calculate the loss function.

2.2.2 Improvement. (1) Introducing a hybrid dilated convolution module to expand the receptive field of the network

The stroke size increases as the size of the style image increases. When the receptive field of a branch in the generation network is smaller than that of its corresponding stroke size, the branch cannot fully draw the stroke and can learn to draw only a portion of the stroke in each region, thus affecting the quality of the style transfer. Therefore, for large stroke sizes, the network needs a larger receptive field to learn the global stroke configuration. To address this problem, we introduce a hybrid dilated convolution module to improve the receptive field of branches with different stroke sizes. The hybrid dilated convolution module is shown in Fig 4. The module comprises three convolution layers with the same convolution kernel size but different expansion rates. The expansion rates of the convolution layers are 1, 2 and 3 in order from front to back. Finally, the output features are added to the input features to obtain the final result.

Compared with that of the ordinary residual block composed of two convolution kernels of 3x3 size, the residual block composed of the dilated convolution module has a larger receptive field. Taking the convolution kernel with a size of $K \times K$ as an example and assuming that the number of channels C remains unchanged, calculating a single pixel in the output feature involves K^*K^*C pixels in the input feature, and the size of the area covered in the spatial dimension is K^*K . The more pixels in the input features involved in a single pixel of the output

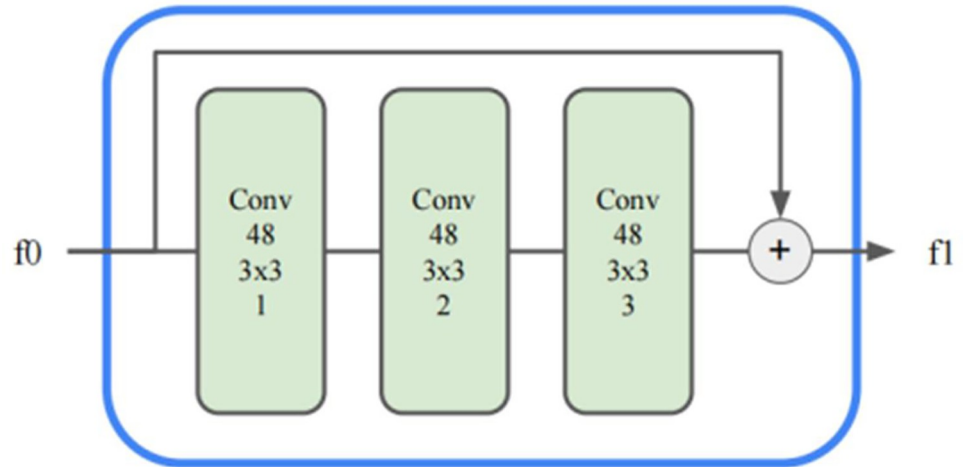


Fig 4. Structure of the hybrid dilated convolution module.

<https://doi.org/10.1371/journal.pone.0284742.g004>

feature, the larger the receptive field of the network. We take the size of the area covered by the associated pixel as the reference for the receptive field size and then compare the receptive field of a single branch before and after the improvement. The number of correlated pixels in the ordinary residual block is $5 \times 5 \times C = 25 \times C$, and the coverage area is 5×5 . The number of associated pixels of the hybrid dilated convolutional residual network is $13 \times 13 \times C = 169 \times C$, and the size of the coverage area is 13×13 . The results show that the hybrid dilated convolution module increases the receptive field of a single branch by more than 6 times the original approach and larger strokes are learned to improve the global effect. Moreover, the dilated convolution of such permutations and combinations makes the coverage more uniform while avoiding the checkerboard effect caused by the reuse of the associated pixels in the spatial dimension.

(2) Introducing the mean standard deviation loss to enhance the network sensitivity to shape features

Conventional style transfer algorithms use the Gram matrix as a method to measure style similarity. However, the Gram matrix is not sensitive to the shape structure in the style image spatial dimension. For the style image with no obvious color difference between adjacent regions, the Gram matrix has difficulty retaining the structural style image information, resulting in a small loss in the Gram matrix during the training process, but the effect of generating the image is very poor. In this paper, we introduce the mean standard deviation loss as an additional loss term in the style loss function. The calculation steps of the mean standard deviation loss are as follows: The style image and the output image are introduced into VGG-Net to extract features, the mean and standard deviation of the features are calculated in the spatial dimension, and the results are connected in the channel dimension. Finally, the MSE loss of the two is calculated.

The mean and standard deviation are calculated as follows:

$$\mu(F) = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W F_{hw} \tag{1}$$

$$\sigma(F) = \left[\frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W (F_{hw} - \mu(F))^2 \right]^{\frac{1}{2}} \tag{2}$$

$$L = \|\mu(F^l(I_o)) - \mu(F^l(I_s))\| + \|\sigma(F^l(I_o)) - \sigma(F^l(I_s))\| \tag{3}$$

where I_o represents the style image, I_s represents the output image, and $F^l(\bullet)$ represents the first layer feature extracted by VGG-Net.

(3) Introducing defocus prediction and fusing multi-stroke features to enhance semantic information in generated images

We use a neural network to predict the defocus map of a content image. The defocus prediction module consists of a 1 channel expanded convolution layer, 7 residual blocks and a reconstruction layer. The channel expansion layer is responsible for expanding the 24-channel features into 48-channel features; the reconstruction layer uses a single convolutional layer and a sigmoid activation function to convert the 48-channel features into a single-channel defocus map. Since the defocus map is used for the fusion operation of the branch features of the stroke pyramid, the size of the defocus map needs to be the same as that of the branch features, and no upsampling operation is needed.

The previous style transfer approach does not use a variety of transfer strategies for the image subject and background, usually rigidly synthesizing a whole image with a single stroke. When the transfer texture adopts a monotonous and regular shape style, the boundary between the subject and the background in the generated image is weakened so that the viewer cannot see the content that the image wants to express. In this paper, we highlight the main area of the image by controlling the relationship between the degree of defocus blur and the stroke size in the spatial dimension. The reason for image defocusing is usually that the object is too far from the lens to focus.

The relationship between the degree of blur and the stroke size is based on the idea that the image clearly expresses itself. The higher the ambiguity of the region and with less semantic information, larger strokes should be used; the lower the ambiguity of the region and with more semantic information, smaller strokes should be used. This stroke fusion strategy can highlight the main area of the image and depict the content of the clear area in more detail. In this paper, the degree of ambiguity is divided into several levels. Different blur-level regions are synthesized using strokes of their corresponding sizes. There is a transition region between the adjacent regions of the fuzzy level, and the stroke size increases with increasing ambiguity in the transition region.

We use the defocus map obtained by defocus blur estimation as the stroke weight map D , and its range is $[0,1]$. Each element represents the size of the stroke in the spatial position, where 0 represents the larger stroke and 1 represents the smaller stroke. We enhance the contrast of the stroke weight map D to obtain the weight relationship between adjacent branches in the spatial dimension.

Let W_i be the weight relationship between the i -th branch and the $i+1$ -th branch. The calculation formula of the weight relationship is:

$$w_{i,j} = \begin{cases} 0, & d_j < u_i \\ \frac{d_j - u_i}{v_i - u_i}, & u_i \leq d_j < v_i \\ 1, & d_j > v_i \end{cases} \tag{4}$$

$i = 1, \dots, n, j = 1, \dots, N$

where $w_{i,j}$ is the j -th weight in W_i and d_j is the j -th weight in the stroke weight graph D . u_i, v_i are the corresponding transition thresholds.

Then, the weight relationship is converted into a set of soft masks, and each branch has its corresponding soft mask. The code value for the soft mask reflects the attention to the corresponding position of the branch feature in the spatial dimension. The calculation formula for the soft mask is as follows:

$$\begin{cases} M_1 = A - W_1 \\ M_i = W_i - W_{i+1}, i = 2, \dots, n - 1 \\ M_n = W_n \end{cases} \tag{5}$$

where M_i is the soft mask corresponding to the i -th branch and A is the full 1-matrix with the same spatial dimension as the depth map.

Finally, the formula to calculate the soft mask gating function G_i of the i -th branch is given as follows:

$$G_i(D, F_{B_i}) = M_i \otimes F_{B_i} \tag{6}$$

where M_i is the soft mask corresponding to the i -th branch, F_{B_i} is the feature map output by branch F_{B_i} , and \otimes represents the dot multiplication operation of the spatial dimension.

The feature maps output by all branches need to go through the soft mask gating function and accumulate. Then, that output goes through the decoder.

$$I_o = Dec \left(\sum_i G_i(D, F_{B_i}) \right) \tag{7}$$

where Dec represents the decoder network, I_o represents the stylized image, and G_i represents the calculated result of the i th branch feature after going through the soft mask gating function.

2.2.3 Loss function calculation. (1) Style transfer loss function

The loss function consists of three parts: content loss, style loss and total variation regularization loss. Content loss is used to preserve the semantic information in the content image. The calculation formula is as follows:

$$L_c = \sum_{l \in \{l_c\}} \|F^l(I_c) - F^l(I_o)\| \tag{8}$$

where $F^l(I_o)$ and $F^l(I_{content})$ are the features extracted by the output image and the content image input to VGG-Net19 [35] in the l layer, respectively; the content extraction layer $\{l_c\}$ is set to $\{conv4_1\}$.

The total variation regularization loss is used to improve the smoothness of the output image, and its implementation is consistent with the implementation method in [15]. Style loss is used to measure the style similarity between the style image and the output image. We use the style loss calculation method proposed in [13]. The calculation formula for the Gram matrix is as follows:

$$G(F^l(I_s)') = [F^l(I_s)'] [F^l(I_s)']^T \tag{9}$$

where $F^l(I_s)' \in R^{C \times (H \times W)}$ is the feature matrix of feature map $F^l(I_s) \in R^{C \times H \times W}$ after the reshaping operation.

Style loss consists of Gram style loss and mean standard deviation loss.

The calculation formula of Gram style loss is as follows:

$$L_{s1} = \sum_{l \in \{I_s\}} \|G(F^l(R(I_s, T_{B_k})))' - G(F^l(I_O^{B_k}))'\| \tag{10}$$

The calculation formula of the mean standard deviation loss is as follows:

$$L_{s2} = \sum_{l \in \{I_s\}} [\|\mu(F^l(R(I_s, T_{B_k}))) - \mu(F^l(I_O^{B_k}))\| + \|\sigma(F^l(R(I_s, T_{B_k}))) - \sigma(F^l(I_O^{B_k}))\|] \tag{11}$$

where I_s represents the style image, $R(I_s, T_{B_k})$ represents the style image scaled to T_{B_k} resolution, and $F^l(R(I_s, T_{B_k}))$ and $F^l(I_O^{B_k})$ represent the corresponding output image of branch B_k and the scaled style image input to the VGG-Net to extract the features in the l layer, respectively.

In this article, we set the style extraction layer with three branches {conv1_2, conv2_2, conv3_3, conv4_3}. The scaling resolution corresponding to each branch is set to x256, x512, x768 respectively.

Finally, the total loss function is:

$$L = \lambda_{s1}L_{s1} + \lambda_{s2}L_{s2} + \lambda_cL_c + \lambda_{tv}L_{tv} \tag{12}$$

where $\lambda_{s1}, \lambda_{s2}, \lambda_c, \lambda_{tv}$ are the superparameters of the Gram style loss, mean and standard deviation loss, content loss and tv loss, which are set as 1e5, 25, 1, 5 and 1e-7, respectively.

(2) Defocusing estimation loss function

The loss function consists of defocus loss and semantic loss.

The defocus loss is calculated from the MSE loss of the defocus predicted by the module and the validation defocus in the dataset. The semantic loss is calculated by the MSE loss of each layer feature extracted by VGG-Net from the original image and the prediction image.

The formula of the loss function is as follows:

$$L_{defocus} = \lambda_1 \|I_{blur} - I_o\| + \lambda_2 \|F^l(I_{blur}) - F^l(I_o)\| \tag{13}$$

where I_{blur} is the verification defocus map in the dataset, I_o is the defocus map output by the defocus estimation module, and $F^l(\bullet)$ is the l layer feature extracted after the image is transmitted to VGG-Net. The extraction layer is {conv1_2, conv2_2, conv3_3, conv4_3}.

2.4 Experiment

2.4.1 Experimental datasets. We use Microsoft’s Common Objects in Context (COCO)-2017 public dataset for image processing [36] to train the style transfer network. This dataset is a large-scale universal dataset for deep learning model training. It contains more than 330,000 images of objects of over 80 categories, and these objects possess abundant context information in real life. These images exhibit objects of different categories, such as people, vehicles, animals and furniture, as well as objects in different situations (e.g., an image may show a person in sportswear playing basketball on the court or a cat resting on the sofa). Compared with other datasets, the images contained in the COCO dataset contain more abundant context information, which makes them more challenging in style transfer tasks for speech information processing and more authoritative in calculating evaluation indicators. The images of the training set are scaled and cropped on the original basis and uniformly converted to 512x512 size. The SYNDOD dataset [34] is used, which contains artificially synthesized blurred images and corresponding defocus maps. The images and defocus maps of the training set are reduced by 4 times on the original basis.

2.4.2 Training details. In this study, we use PyTorch to build the model and train it on a GeForce RTX 2080 Ti graphics card. The normalization operation is performed uniformly before the image is input into the feature extraction network. The averages of the input images are adjusted to [0.485, 0.456, 0.406], and the standard errors are adjusted to [0.229, 0.224, 0.225]. The number of samples contained in a single training batch is set to 2. For the dropout setting, we randomly set some feature values to 0 with a probability of 0.5 during training and set the skip threshold to $1e5$ to ensure a skip over the current batch of training when the loss value fluctuates largely. In this study, the Adam optimizer is used for training, with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$, and $\varepsilon = 1e-8$.

The training in this paper is divided into two parts: the defocus estimation module and the stroke pyramid training.

For stroke pyramid training, the COCO-2017 dataset is used, and the learning rate is set to $1e-3$. Since the training set is large enough, only 1 epoch is trained, and the learning rate attenuation is not set. To train the various stroke branches in a network, we adopt a progressive training strategy. Each training batch selects only one branch for training, and we poll the branch index so that the parameters of each branch can be updated. In addition, under the progressive training strategy, the training of the latter stroke branch benefits from the knowledge of the previously learned branch.

For defocus estimation module training, the SYNDOF depth of the field dataset is used for training, and the learning rate is set to $1e-4$, which gradually decays to $1e-5$ as the training epoch increases, with a total of 60 epochs trained.

2.4.3 Experimental analysis. To better verify the use value of the model in this paper, we compare the model with other classical algorithms. From the training curve analysis, the verification of the effect of the residual block of the mixed cavity convolution, the comparison of the loss effect of the mean standard deviation, the comparison of the defocus adaptive and the single stroke, and ablation experiments were conducted in four parts.

3. Results and discussion

3.1 Contrast experiment

To verify the effectiveness of the proposed method in this paper, the mainstream image style transfer methods AdaIn [22], WCT [10], AAMS [25] and model stroke pyramid [24] are used for effect comparison. The purpose of this method is to keep the content expression of the subject as much as possible when dealing with shallow depth of field images so that it does not integrate with the texture generated by the background. As shown in Fig 5, the main contour of the image generated by AdaIn is clear, but the color display is different from the style image. The image generated by WCT is better in color conversion, but the boundary between the subject and the background is not obvious; moreover, it is not easy to express its semantic information. The image generated by the method in this paper maintains a good semantic expression in a clear area, and the texture in the blurred area is rough so that it can more clearly show the content of the image focus. For example, in the first photo of a teddy bear, the teddy bear is the main content, which is the focus area of the image. In addition, behind the teddy bear is the background content, which is the defocused area of the image. After the content image is migrated, the gray-yellow lattice style is obtained. The teddy bear area is expressed by a smaller lattice, and its detailed texture is displayed. The background area is expressed by a larger lattice to distinguish it from the main content, highlighting the semantic expression of the main content, and the transition between the main area and the background area is continuous and natural. Compared with those of the baseline model, the strokes of the images generated by the algorithm are more complete, and the shapes and lines are more

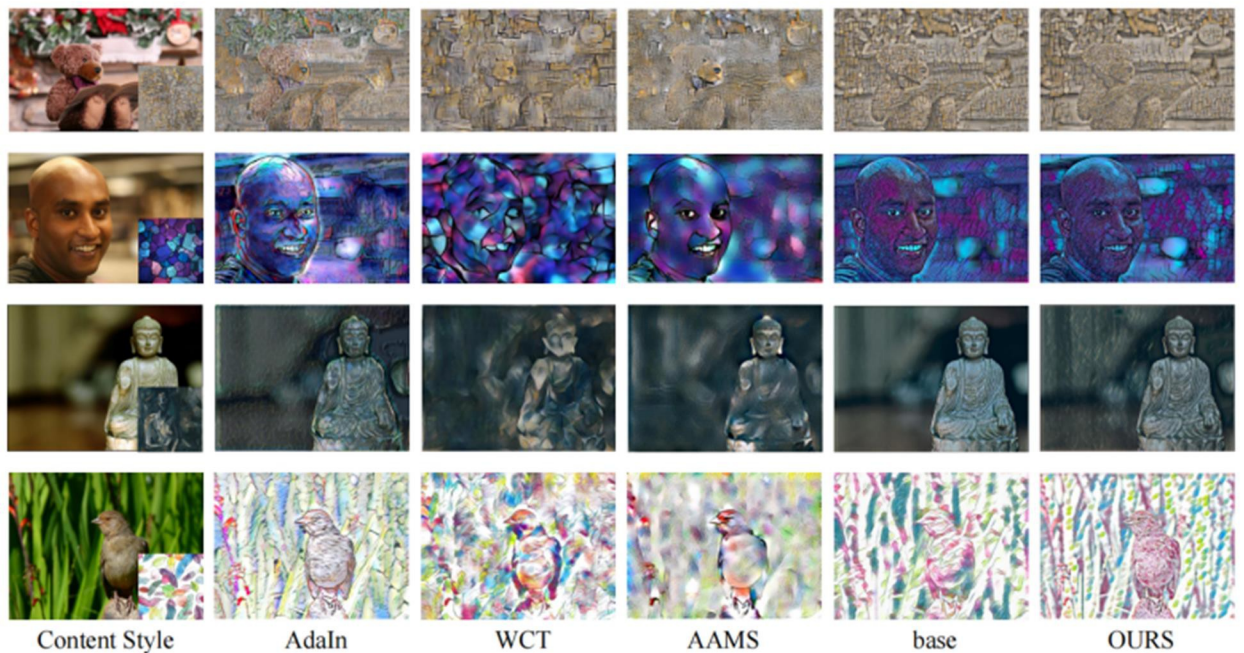


Fig 5. Comparison of results generated by the various methods.

<https://doi.org/10.1371/journal.pone.0284742.g005>

similar to the style images. As in the second image, the image generated in this paper fully expresses the square block pattern, while the baseline model cannot describe the large strokes due to the lack of receptive field; thus, the purple color spots appear in the face area.

Since the style transfer results are difficult to evaluate by quantitative indicators, we use the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) to evaluate the antinoise ability of this method and the ability to maintain the semantic content images. In this paper, various methods are analyzed on the test set. The evaluation index uses the generated image and the content image to calculate the average value. The higher the two indices are, the better. As shown in Table 2, the proposed method has certain advantages in texture clarity and semantic information retention.

3.2 Ablation experiment

3.2.1 Training curve comparison. To analyze the descent process of the loss function, we compared the training process of the baseline model network with the network with mixed dilated convolution. A test was performed on 4,000 samples per training, and the style loss and content loss of the results generated by the three branches were recorded. The recorded values were calculated as the average of the loss values of 4 style images and 100 content images. As

Table 2. Comparison of quantitative indices between the various methods.

	AdaIn		WCT		AAMS		base		OURS	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
style image composition	11.9	0.31	11.75	0.34	11.67	0.31	11.96	0.34	12.05	0.36
cubes	14.65	0.5	11.6	0.41	13.99	0.53	12.73	0.58	12.78	0.59
women	14.23	0.54	11.92	0.43	14.36	0.62	14.59	0.64	14.62	0.67
leaves	7.25	0.33	6.91	0.23	7.45	0.37	7.34	0.3	7.36	0.3

<https://doi.org/10.1371/journal.pone.0284742.t002>

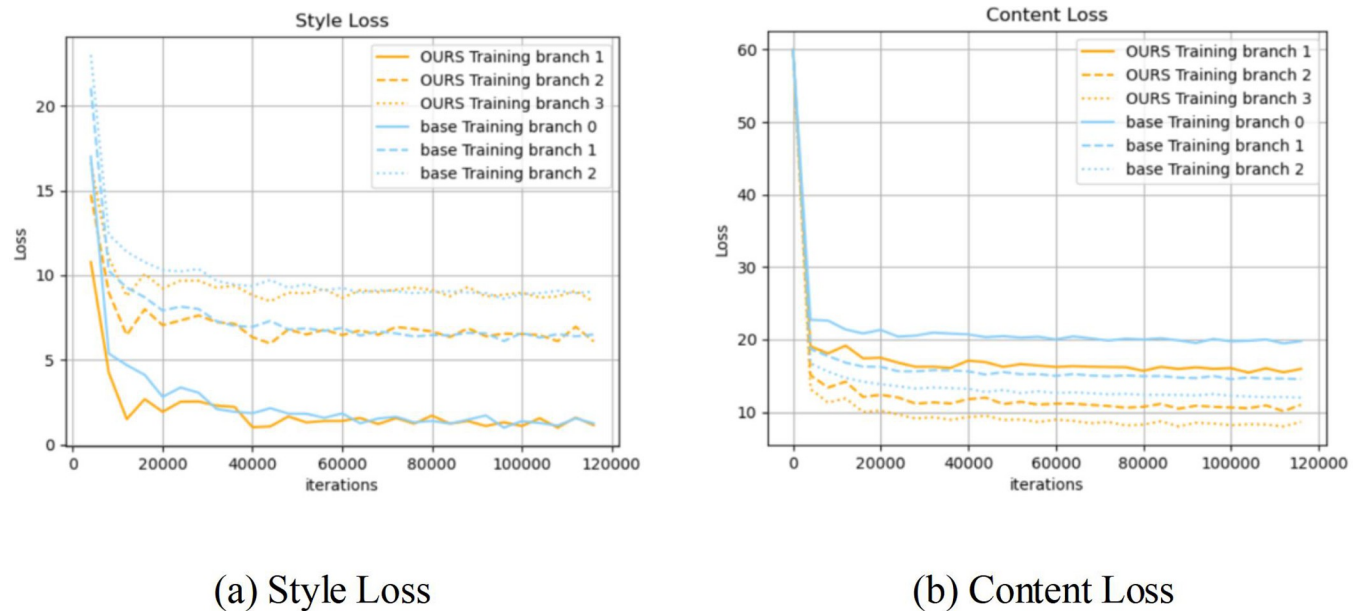


Fig 6. Comparison of the training curve between the network when using hybrid dilated convolution and the baseline model network.

<https://doi.org/10.1371/journal.pone.0284742.g006>

shown in Fig 6, the style loss value calculated by each branch of the network with mixed dilated convolution decreased rapidly in the early stage of training, and the content loss was slightly lower than that of the baseline model, which indicates that the network training speed had a certain degree of improvement.

3.2.2 Ablation experiment for the cavity convolution. Cavity convolution can expand the receptive field of the network so that the network can learn to depict larger strokes. To verify the effect of cavity convolution in this model, we selected several style images with larger strokes and compared the results of the network output by using ordinary residual blocks and the network output by using mixed cavity convolution residual blocks. To verify the ability of independent branches to depict strokes, we do not fuse the features of each branch and instead use the independent output results of each branch.

As shown in Fig 7, the image output by the network model when using the ordinary residual block is more complete in the strokes of the first two branches, while the last branch has no obvious shape features and does not draw a larger square block pattern. It can describe only the color and some lines of the style image; there are gradual color spots. The image output by the network model when using the mixed hole residual block can draw a large stroke well in the last branch, and the drawn block is more three-dimensional in the results generated by each branch.

3.3.3 Ablation experiment for the mean standard deviation loss. To verify the influence of the mean standard deviation loss on the model in this paper, a style image with darker color was selected for comparative experiments. The network trained with mean standard deviation loss was compared with the network trained without mean standard deviation loss.

As shown in Fig 8, the image generated by using only the Gram matrix to calculate style similarity cannot effectively transfer the colorless style, and the effect is similar to color transfer. However, by introducing the mean standard deviation loss, the network can retain some structural information in the style image to a certain extent.

3.3.4 Defocus adaptive ablation experiment. We compared the output results of the multi-stroke fused with defocus images approach with the results of the single stroke approach.

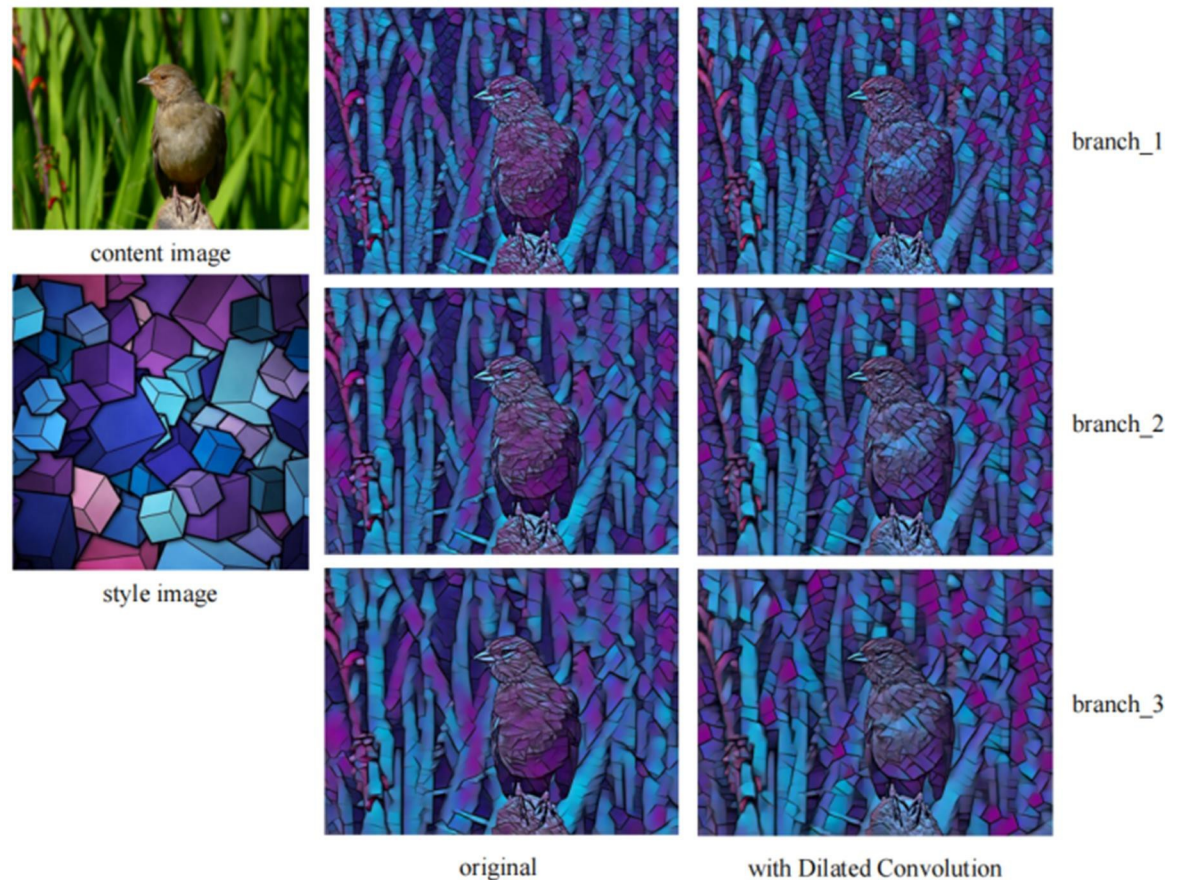


Fig 7. Comparison of the effect before and after the introduction of cavity convolution.

<https://doi.org/10.1371/journal.pone.0284742.g007>

As shown in Fig 9, the use of multi-stroke fused with defocus images produces a three-dimensional effect. The texture displayed as the main area is more delicate and has smaller strokes. The objects displayed as the background area are depicted with larger strokes, and the painting style is bolder to highlight the semantic expression of the main area.

3.3.5 Time performance test. To test the time performance of the framework proposed in this study, we reduce the size of the test samples from the COCO-2017 dataset, which contains 10,000 test samples, to 256, 512 and 1024, respectively, and then input them into the model. We calculate the time consumption of the style transfer module and defocus estimation module as well as the total time consumption of the entire framework. Table 3 summarizes the average time consumption of the module with seconds as the unit. Compared to the classical method proposed by Johnson et al. [15], our method uses convolutional layers with smaller channels, which can process large-size images faster. Although the processing speed of the style migration module involved in our study is slower than the baseline model [24] as a consequence of the increased network depth, the performance loss is within an acceptable range.

4. Conclusion

In this paper, an improved multi-stroke defocus adaptive style transfer based on a stroke pyramid is proposed. This method addresses the problem that image style transfer causes a certain loss of semantic information of the original image after giving the image a specific style. By



Fig 8. Comparison of effects before and after the introduction of mean standard deviation loss.

<https://doi.org/10.1371/journal.pone.0284742.g008>

introducing the stroke pyramid, the one-stop generation of stylized images with multiple stroke sizes is realized. By introducing the mixed dilated convolution, the receptive field of each branch network is expanded so that the network can learn the larger geometric structure features in the style image. By using the feedforward network to generate the defocus

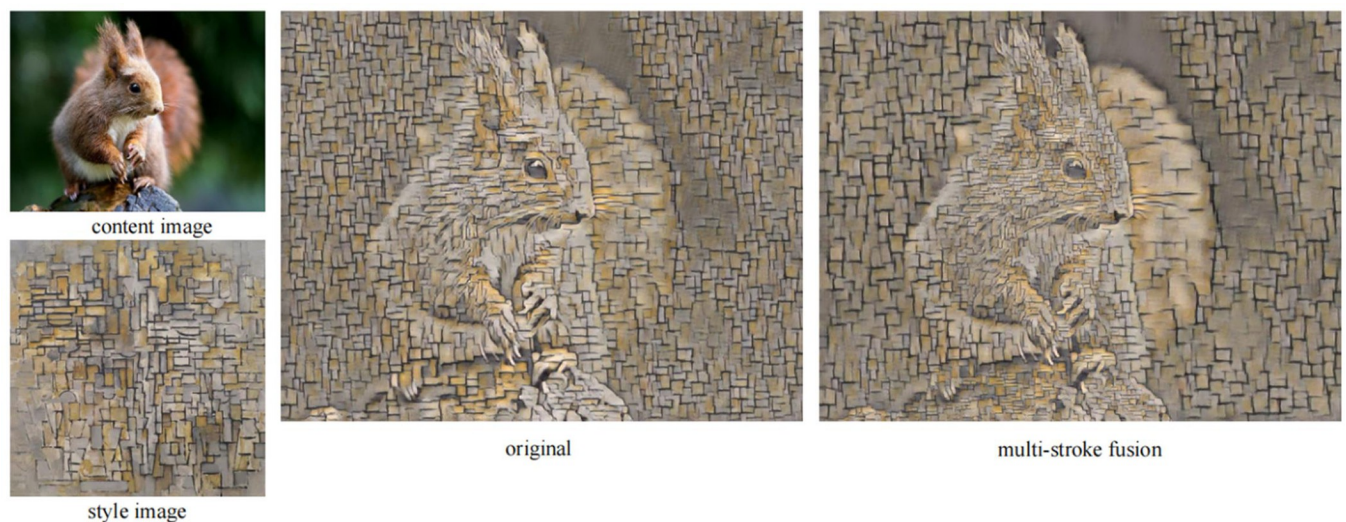


Fig 9. Comparison of the effects before and after the introduction of defocus adaptive multi-stroke feature fusion.

<https://doi.org/10.1371/journal.pone.0284742.g009>

Table 3. Time performance table of each module under different image sizes (unit, second).

Image size	Style transfer module	Defocus estimation module	Total
256x256	0.0207	0.01194	0.0368
512x512	0.0234	0.01291	0.0398
1024x1024	0.0243	0.01495	0.0448

<https://doi.org/10.1371/journal.pone.0284742.t003>

estimation map of the image in the multibranch fusion, the mixed expression of multiple strokes in the spatial dimension of the image is realized. The experimental results show that compared with that of the classical style transfer method, the texture of the image generated by our framework is more delicate in the clear area, the texture in the fuzzy area is softer, and the difference between the main area and the background area of the image is more significant. This approach makes it easier for viewers to understand the semantic content expressed by stylized images, and the PSNR and SSIM indicators are improved by 1.43 and 0.12 on average, respectively. Although the processing effect of the framework is better in subjective evaluation, the defocus estimation module uses fewer convolutional layers, and the defocus estimation is not accurate enough, so some missing image blocks are generated.

In the future, we will consider increasing the number of residual blocks in the defocus estimation module to 20, and we will add an attention mechanism to further improve image quality. Through these treatments, the model complexity can be increased, which enables the model to be more suitable for processing high-resolution images and complex scenes. In addition, we plan to increase the number of branches in the style migration network to 5 to obtain richer stroke and texture information and then determine whether the final effect can meet expectations according to the assessment of the results. For future prospects, the structure of arbitrary style transfer networks and multi-stroke style transfer may constitute one of the future development directions. These two methods can improve the flexibility of the network while maintaining high-quality image generation, allowing users to choose more styles and styles to generate images. In addition, these methods can greatly shorten the generation time and model size, thereby better adapting to practical application scenarios.

Author Contributions

Conceptualization: Zeyu Chen.

Data curation: Jianfang Cao.

Formal analysis: Zeyu Chen.

Funding acquisition: Jianfang Cao.

Investigation: Zeyu Chen, Yun Tian.

Methodology: Jianfang Cao, Yun Tian.

Project administration: Jianfang Cao, Yun Tian.

Resources: Yun Tian.

Software: Mengyan Jin.

Supervision: Mengyan Jin.

Validation: Mengyan Jin.

Visualization: Mengyan Jin.

Writing – original draft: Zeyu Chen, Yun Tian.

Writing – review & editing: Jianfang Cao.

References

1. Haq MA, Jilani AK, Prabu P. Deep Learning based modeling of groundwater storage change. *Computers, Materials & Continua* 2022; 70(3): 4599–4617.
2. Haq MA. CDLSTM: A novel model for climate change forecasting. *Computers, Materials and Continua* 2021; 71(2): 2363–2381.
3. Haq MA, Ahmed A, Khan I, Gyani J. Analysis of environmental factors using AI and ML methods. *Scientific Reports* 2022; 12(1): 13267. <https://doi.org/10.1038/s41598-022-16665-7> PMID: 35918395
4. Haq MA. SMOTEDNN: A novel model for air pollution forecasting and AQI classification. *Computers, Materials and Continua* 2021; 71(1): 1403–1425.
5. Haq MA. Planetscope nanosatellites image classification using machine learning. *Computer Systems Science and Engineering* 2022; 42(3): 1031–1046.
6. Haq MA. CNN based automated weed detection system using UAV imagery. *Computer Systems Science and Engineering* 2022; 42(2): 837–849.
7. Haq MA, Rahaman G, Baral P, Ghosh A. Deep learning based supervised image classification using UAV images for forest areas classification. *Journal of the Indian Society of Remote Sensing* 2020; 49(3): 601–606.
8. Haq MA. Intelligent sustainable agricultural water practice using multi sensor spatiotemporal evolution. *Environmental Technology* 2021: 1–14. <https://doi.org/10.1080/09593330.2021.2005151> PMID: 34842040
9. Patro KK, Prakash AJ, Hammad M, Tadeusiewicz R. SCovNet: A skip connection-based Feature union deep learning technique with statistical approach analysis for the detection of COVID-19. *Biocybernetics and Biomedical Engineering* 2023; 43(1): 352–368. <https://doi.org/10.1016/j.bbe.2023.01.005> PMID: 36819118
10. Prakash AJ, Patro KK, Hammad M, Tadeusiewicz R. BAED: A secured biometric authentication system using ECG signal based on deep learning techniques. *Biocybernetics and Biomedical Engineering* 2022; 42(4): 1081–1093.
11. Patro KK, Prakash AJ, Samantray S, Pławiak J. A hybrid approach of a deep learning technique for real-time ECG beat detection. *International Journal of Applied Mathematics and Computer Science* 2022; 32(3): 455–465.
12. Patro KK, Prakash AJ, Rao MJ, Kumar PR. An efficient optimized feature selection with machine learning approach for ECG biometric recognition. *IETE Journal of Research* 2020; 68(1): 1–12.
13. Gatys LA, Ecker AS, Bethge M. Image style transfer using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, 2414–2423.
14. Fan Q, Chen D, Yuan L, Hua G, Yu N, Chen B. Decouple learning for parameterized image operators. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018: 442–458.
15. Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution. In: *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*. Springer International Publishing, 2016: 694–711.
16. Ulyanov D, Lebedev V, Vedaldi A, Lempitsky V. Texture networks: Feed-forward synthesis of textures and stylized images. 2016; arXiv preprint arXiv:1603.03417.
17. Zhang H, Dana K. Multi-style generative network for real-time transfer. 2017; arXiv preprint arXiv:1703.06953.
18. He M, Chen D, Liao J, Sander PV, Yuan L. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*. 2018; 37(4):1–16.
19. Li Y, Chen F, Yang J, Wang Z, Lu X, Yang MH. Diversified texture synthesis with feed-forward networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 3920–3928.
20. Chen TQ, Schmidt M. Fast patch-based style transfer of arbitrary style. 2016; arXiv preprint arXiv:1612.04337.
21. Huang V, Belongie SJ. Arbitrary style transfer in realtime with adaptive instance normalization. In: *Proceedings of the IEEE international conference on computer vision*. 2017: 1501–1510.
22. Li Y, Fang C, Yang J, Wang Z, Lu X, Yang MH. Universal style transfer via feature transforms. *Advances in Neural Information Processing Systems* 2017; 30.
23. Park DY, Lee KH. Arbitrary style transfer with style-attentional networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019; 5880–5888.

24. Jing Y, Liu Y, Yang Y, Feng Z, Yu Y, Tao D, et al. Stroke controllable fast style transfer with adaptive receptive fields. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018; 238–254.
25. Yao Y, Ren JQ, Xie XS, Liu WD, Liu YJ, Wang J. Attention-aware multi-stroke Style Transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019; 1467–1475.
26. Zhang CY, Yang J, Wang L, Dai ZY. S2WAT: Image Style Transfer via Hierarchical Vision Transformer using Strips Window Attention. Computer Science 2022. <https://doi.org/10.48550/arXiv.2210.12381>
27. Feng JX, Zhang G, Li XH, Ding YM. A compositional transformer based autoencoder for image style transfer. Electronics 2023; 12(5): 1184.
28. Zhuo S, Sim T. Defocus map estimation from a single image. Pattern Recognition. 2011; 44(9):1852–1858.
29. Kraus M, Strengert M. Depth-of-field rendering by pyramidal image processing. In: Computer graphics forum. Oxford, UK: Blackwell Publishing Ltd, 2007; 26(3):645–654.
30. Shi J, Xu L, Jia J. Discriminative blur detection features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 2965–2972.
31. Tang C, Hou C, Song Z. Defocus map estimation from a single image via spectrum contrast. Optics letters. 2013; 38(10):1706–1708. <https://doi.org/10.1364/OL.38.001706> PMID: 23938918
32. Xu G, Quan Y, Ji H. Estimating defocus blur via rank of local patches. In: Proceedings of the IEEE International Conference on Computer Vision. 2017: 5371–5379.
33. Park J, Tai YW, Cho D, Kweon IS. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017; 1736–1745.
34. Lee J, Lee S, Cho S, Lee S. Deep Defocus Map Estimation using Domain Adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019; 12222–12230.
35. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014; arXiv:1409.1556.
36. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: Common objects in context. In: Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014; 740–755.