electronics

MDPI

Article

Adaptive Multilevel Coloring and Significant Texture Selecting for Automatic Deep Learning Image Transfer

Hsien-Chu Wu [1], Yu-Chi Liu [1], Yen-Yu Chen [2,*] and Yu-Yen Weng [3]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan
[2] Department of Artificial Intelligence and Computer Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan
[3] Department of Management Information Systems, National Chung Hsing University, Taichung 40227, Taiwan
* Correspondence: vpcyy2233@gmail.com; Tel.: +886-922638077

Abstract: This paper proposes an image style transfer technique based on target image color and style, which improves the limitations of previous studies that only consider inter-image color transfer and use only deep learning for style transfer. First, an adaptive multilevel cut is made based on the luminance distribution of the two image pixels, and then a color transfer is applied to each region. Next, deep learning is used to select effective features for the target image, and the convolutional layer determines the extent of effective features by using the structural similarity index (SSIM) and black blocks. Selecting a convolutional layer with more effective features can reduce the limitations of the deep learning style transfer that requires artificial control parameters. The proposed method improves image quality by automatically simulating the color and style of the target image and controlling the parameters without human intervention. By evaluating the similarity between the result image and the target image, the proposed method can reduce the gap of variance by more than two times, and the result image can display a balance between the color and style of the target image.

Keywords: deep learning; image style transfer; color transfer; structural similarity index

Citation: Wu, H.-C.; Liu, Y.-C.; Chen, Y.-Y.; Weng, Y.-Y. Adaptive Multilevel Coloring and Significant Texture Selecting for Automatic Deep Learning Image Transfer. *Electronics* **2022**, *11*, 3750. https://doi.org/10.3390/electronics11223750

Academic Editor: Chiman Kwan

Received: 14 October 2022
Accepted: 10 November 2022
Published: 15 November 2022

1. Introduction

With the rapid development of information technology, artificial intelligence (AI) technology is gradually attracting the attention of the industry. Because of the popularity of cell phones, many different applications can be downloaded and installed through online stores. There are many different image processing applications available on the online market. Users can select an original image and choose the desired style, such as hand painting or impressionist painting, to convert the image. After applying the processing, the original image is automatically transformed into a new style image with the desired style. An ideal style transfer technique can serve the scene of the original image in advance and then combine it with the color and style of the target image. How to consider both the color and style of the target image has become an important issue today. P. P. Galanter et al. [1] proposed a definition of generative art where a machine can perform an automatic computation through an algorithm or a set of rules and can eventually generate a work of art. Image transfer techniques can be divided into two parts: image color transfer and image style transfer. Jimenez-Arredondo et al. [2] summarized the following three main approaches to image color transfer: geometry-based methods, user-assisted schemes, and statistical methods. First, when similar features appear on two images, the geometry-based method [3] can automatically search for features that correspond to each other, allowing the corresponding features to be color transferred, and finally creating similar features with similar colors. This method is widely applicable to real-world images. If the structures of the content image and the target image are very different, the geometry-based color transfer method cannot successfully find the

corresponding features, so these images are not suitable for color transfer. Second, the user-assisted solution [4] requires manual user intervention. The user needs to mark the corresponding feature positions of the content image and the target image and then perform the color transfer. This approach can provide better results in terms of the color transfer of features and is less prone to color transfer errors between features. However, the disadvantage of the user-assisted solution method is that if the color distribution of the structure of the content image is more detailed than that of the target image, the user may need to point out the locations of more corresponding features. Although the user-assisted solution offers better results for image color transfer, it also means that this approach is not efficient. Third, the statistical method [5–7] is applicable to the case where the features of the content image and the target image do not directly correspond to each other. Reinhard et al. [5] used a simple statistical analysis to impose one image's color characteristics on another. It can achieve color correction by choosing an appropriate source image and applying its characteristics to another image. Papadakis, N. et al. [6] proposed a variational formulation for the intermediate color histogram equalization of two or more images. Cepeda-Negrete, J. et al. [7] used a classical color transfer method to obtain first-order statistics from a target image and transfer them to a dark input, modifying its hue and brightness. The statistical method calculates the ratio of pixel values between the content image and the target image and then performs the image color transfer. The results of the image color transfer are better if the two images have similar structures.

The scales of the image color transfer can be divided into two types: global color transfer and local color transfer. Reinhard et al. [5] proposed a statistically based method for image color transfer, which is a global color transfer method. This method transforms the image into a suitable color space beforehand and then adjusts the image color by averaging and performing standard deviation on the target image and content image for each pixel. This image color transfer by statistical method is very fast. However, the following limitations may occur. If the color of the target image is more diverse, the color transfer result of the content image will be unnatural. Liu et al. [8] proposed a method to allow users to actively select regions that can correspond to two images for the color transfer between images. By interacting with the user, the region of interest is selected for the color transfer between the target image and the content image. Therefore, owing to the relative abundance of image colors, much human effort is required for manual marking. Khan et al. [9] proposed that multiple target images be used to perform image color transfer. This is a local color transfer algorithm between multiple images based on simple statistics and local linear embedding for edit propagation. The color features of multiple target images are fused into the content image by the user's labeling and statistical methods for the corresponding images. In this approach, the closer the structure of the content image and the target image (sky and sky, sea level and sea level, or flower and flower), the more effective the color transfer will be. As can be seen, the limitation of image color transfer is that it cannot simulate the texture pattern of the target image. Liu et al. [10] proposed an emotional color transfer method for texture perception of images or target images based on emotional words. This is a local color transfer method, so the content image is segmented, and the primary color of the texture is extracted. This method considers image segmentation methods in terms of object segmentation and primary color extraction, which can solve the problem of the same block being assigned to different target colors.

However, the above image color transfer method can simulate only the color technique of the target image but not the artist's brushstroke pattern or the style of the target image. Jimenez-Arredondo et al. [2] proposed an image style transfer method that considers both image color transfer and image style transfer. In this method, the color transfer is first performed between the content image and the target image, and then it is merged with the canvas image that resembles the style of the target image to express the Fauvism. Since this method uses any image to mimic the texture of the target image, the texture of the image generated by this method is different from the texture of the target image. Gatys et al. [11] proposed a deep learning-based image style transfer method. The Gram matrix is used

to characterize the image, and the features of the content image and the target image are combined to produce a new style image. During the image style transfer, the colors and textures on the image are transferred together. The method proposed by NVIDIA and X. Li et al. [12] applies linear style transformation and two small convolutional neural networks. This method is relatively computationally efficient and can save computational costs compared to algorithms that require GPU computation. In the method of X. Li et al. it was found that the style-converted image could keep the edges of the content image, but the style in the target image was less obvious. Since deep learning can extract the features of the image [13], the feature maps of each convolutional layer can be displayed by filters. There are many types of filters in addition to edge detection and color detection.

Liu et al. [14] used pre-training VGG19 in the encoder to capture the features of the style image and content image, and then fused the features with AdaIN. To obtain the edge feature from the content image, a refine network is designed to enhance the image edges by using a modified edge detection network, HED. Because the encoder only focuses on the global feature but ignores the detailed part, the result generated by the transform module has blurred image edges. The disadvantage of this approach is that maxpooling selects features from the feature map and passes them to the next layer, which can filter features and reduce the number of parameters but may also lose other important features. In addition, the VGG19 architecture focuses on global features, and as the network deepens, the texture features of the previous layer may not be passed to the next layer. Because the style transfer focuses on the texture, i.e., the lower-level features in the previous layers, it may lead to obtaining under-styled generated images. In Hung et al.'s proposed method [15], the content image and reference image are used in the semantic match stage to extract the image features by VGG19, and the cosine similarity matrix is calculated to generate the semantic-assisted image. The generator of the translation network uses an autoencoder structure, including one encoder, two decoders (Task 1, Task 2). Task 1 is used to generate the image in the target domain, the U-net design is used in the decoder to enhance the image features and avoid gradient disappearance. Task 2 is used to output the image segmentation map as an attention mask, which allows the model to learn information about the attention regions. Finally, a discriminator is used to determine the generated image. The drawback of Hung et al.'s method is that the semantic match stage calculates the feature similarity between content image and reference image to preserve the spatial and semantic similarities and to match them, while VGG19 lacks multi-scale feature fusion and the feature map of each scale is not operated, which may lead to the lack of detailed features in semantic messages. VGG19 lacks multi-scale feature fusion, and the feature maps at each scale are not manipulated, which may lead to the lack of detailed features in the semantic information. Because the lower-level features have high resolution, they can contain more location and detail information, but less semantic information. The higher-level features have stronger semantic information, but lower resolution and poorer performance for details. Combining the two retrieved features can provide a better match between the color and style of the target image. The proposed method of image color transfer is a type of local color transfer; therefore, it can improve the result of using global color transfer. By selecting effective features in the convolutional layer, there is no need for manual adjustment to achieve a good visual effect. Since the proposed method improves image quality by automatically simulating the distribution of the color and style of the target image and controlling the parameters without human intervention, image transfer results can be balanced between the color and style of the target image.

In this paper, we propose a local color transfer method for the content image and the target image. First, an adaptive multilevel cut is performed based on the luminance distribution of the two image pixels, and then color transfer is performed for each region. Next, deep learning is used to select the effective features of the target, and the degree of effective features of the convolutional layer is judged by the structural similarity index (SSIM) and black blocks. The selection of convolutional layers with more effective features reduces the limitations of deep learning-based transfer that require manual control of

parameters. The experimental results show that the proposed method can improve the image quality by automatically simulating the color and style of the target image and controlling the parameters without manual intervention.

This paper is organized as follows. Section 2 reviews related works. Section 3 introduces the proposed method. Section 4 shows the experimental results of the proposed method. Finally, Section 5 presents conclusions.

2. Related Works

The image color transfer method based on the statistical method is discussed in Section 2.1, and the method of image style transfer based on deep learning is discussed in Section 2.2. Although the above methods provide good image transfer quality in some types of images, there are still some restrictions.

2.1. Reinhard et al.'s Method

Reinhard et al. [5] proposed an image color transfer based on the statistical method. This method involves global color transfer, the easiest and fastest one. This method maps the mean and the standard deviation of the statistical method based on the pixels of the content image and the target image. The advantage of this method is that the color transfer process of the image is fast. On the other hand, the disadvantage is that only the color transfer can be performed, and the texture of the target image cannot be highlighted. In addition, when the image of this method is restricted to cases where the image structure is relatively close, then the image color transfer is better. Furthermore, if the color richness of the target image is high, then the result of color transfer will be poor. Generally speaking, the RGB color space of an image does not adjust the color change very well. Adjusting the value of RGB will color-shift the whole image. Using RGB to describe color is not intuitive to humans. In addition, the three channels of RGB have strong correlations, and it is difficult to change the color while the three channels are changed properly. Both of the input images are converted from RGB color space to the $l\alpha\beta$ color space defined by Ruderman et al. [16]. In $l\alpha\beta$ color space, changing any channel does not affect other channels, so that the effect of the original image can be better maintained. Thus, channel l represents an achromatic channel, channel α represents a yellow–blue channel, and channel β represents a red–green channel. The mean and standard deviations of the content image are calculated in these three channels, and the mean and standard deviations of the target image are calculated in these three channels as well.

Each pixel of the content image subtracts the mean of the corresponding channel and then multiplies the standard deviation ratio of the target image and the content image. Finally, it adds the mean of the target image, as shown in Equation (1). Figure 1 displays the architecture of Reinhard et al.'s method.

$$O_k = \frac{\sigma_t^k}{\sigma_c^k}\left(c^k - \mu\left(c^k\right)\right) + \mu\left(t^k\right) = \{l, \alpha, \beta\}, \quad (1)$$

where content image c and target image t are input images. σ is the standard deviation in $l\alpha\beta$ channels. Then, the new $l\alpha\beta$ color space result is converted to the RGB space.

2.2. Gatys et al.'s Method

In recent years, many researchers have proposed several classic models based on the convolutional neural network (CNN), such as LeNet [17], AlexNet [18], and VGGNet [19], which have achieved good recognition results in the field of image recognition. The VGG19 model was proposed by the Visual Geometry Group at Oxford and Google DeepMind company researchers. Gatys et al. [11] used the VGG19 model with weights pre-trained on the ImageNet dataset [20] to perform an image style transfer. The texture was synthesized by making a Gram matrix according to the corresponding convolutional layer. This method matched the style features in layers *con1_1*, *conv2_1*, *conv3_1*, *conv4_1*, and *conv5_1* as well as matched the content in layer *conv4_2*. In addition, this method separately calculated

the content and style of the image, computed the style loss function and the content loss function, and then combined them to obtain the total loss. If the ratio α/β were 10^{-3} or 10^{-4}, there would be better resulting images.

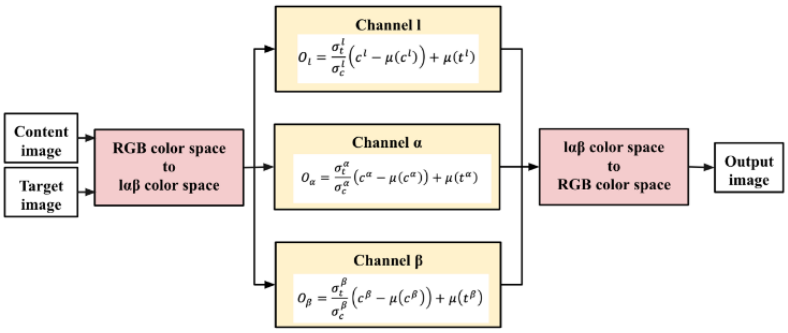


Figure 1. The architecture of Reinhard et al.'s method.

Through the gradient descent (L-BFGS), in the continuous iteration, the resulting image conforming to the loss function description is generated. The process of deep learning style transfer also includes color and texture styles to generate new output images.

Figure 2 shows the architecture of the image style transfer method. However, the disadvantage of Gatys et al.'s method is that manual adjustment of parameters is required. Different images must be manually re-adjusted to find the best image style transfer result. Furthermore, it does not automatically find the best resulting image. Therefore, different images will have different parameter combinations. If the user wants to transfer the style of multiple images, the system operation will be inefficient. The images in this method have 512×512 pixels, and the procedure could take up to an hour on an NVidia K40 GPU.

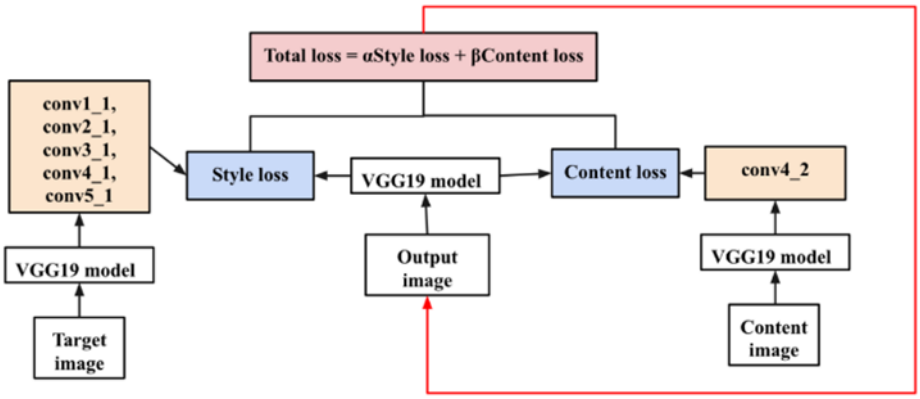


Figure 2. The architecture of image style transfer.

3. The Proposed Method

An ideal image transfer technology transfers the color and style of the target image to the content image. In the proposed method, two images are taken: an original image (referred to as a content image) and an image that will be transferred (referred to as a target image, such as an artwork made by a famous painter). The target image contains two important components: color and style. Color can be used to express the emotion of the painter, and the style of the image can show the painting technique of the painter. As shown in Figure 3, the proposed framework has two main steps. In the first step, after undergoing the color transfer phase (the red line in Figure 3), there will be an output image, Figure 3c. In the second step, the target image, Figure 3b, will be presented, and the content image, Figure 3c, will perform the color transfer; through the style transfer (green line in Figure 3), the new image will be the resulting image, Figure 3d.

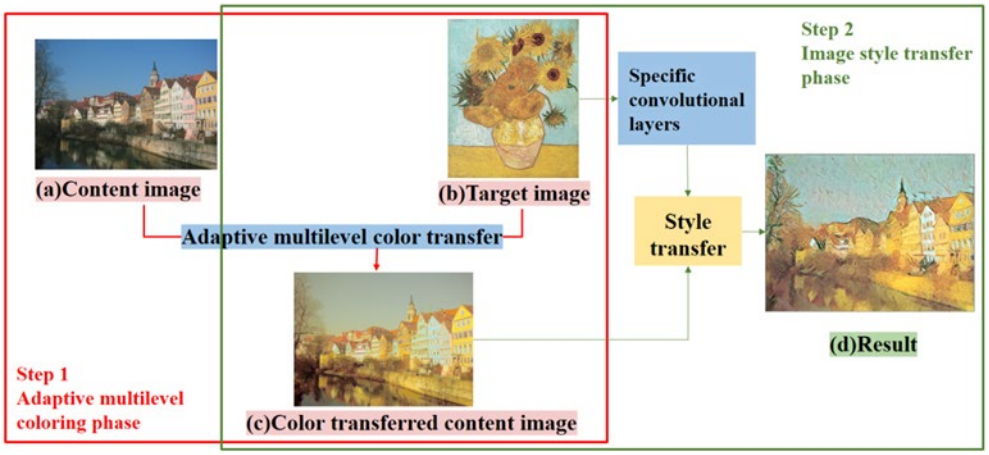


Figure 3. Overview of the proposed framework.

3.1. Adaptive Multilevel Coloring Scheme

In this section, performing the adaptive multilevel color transfer of an image is described. The first phase explains how the proposed method performs adaptive multilevel settings based on the image, and the adaptive multilevel limits between the content image and the target image. The second phase describes how to perform the color transfer after the adaptive multilevel settings are completed for both images.

3.1.1. Adaptive Multilevel Setting

In addition to the color of an image, the luminance distribution of the image is an important factor. In this phase, the proposed method brings both the target image and content image into the chromatic channel of the CIELAB color space [21–23]. The CIELAB color space was defined by the International Commission on Illumination (CIE). CIELAB is known for its perceptual uniformity, and its L component more closely matches human perception of lightness than other color spaces. Otsu's thresholding [24] can divide the image into foreground and background. In this proposed method, the image is divided into multiple foregrounds and backgrounds first, and then the image color transfer is performed. According to the luminance of the target image, this paper used several thresholds to split the target image and content image into multiple regions. In Figure 4, after using the content image and the target image as inputs to perform the adaptive multilevel color transfer, the color of the content image is similar to that of the target image.

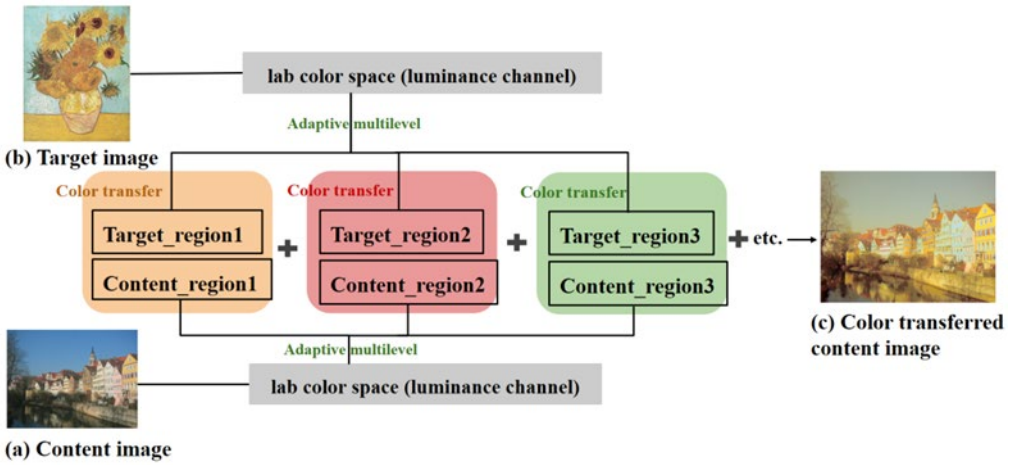


Figure 4. The adaptive multilevel color transfer phase.

With the adaptive multilevel coloring scheme, the image is split into several regions. The range of the L component of the image is $[0, 255]$. The range is represented as a_i and b_i,

and $i \in \{0, 1, \ldots\}$. Here, if $= 0$, then a_0 and b_0 are used to represent 0 and 255, respectively. These two values are the initial values. If the pixel values of the image are between a_i and b_i, then the mean μ_{a_i,b_i} and the standard deviation σ_{a_i,b_i} of this region will be received. In this region, the luminance level l of the image is regarded as $[l_0, l_1, \ldots, l_{255}]$. According to each luminance level of l, the total number of each luminance level will be $t(l)$. The mean μ_{a_i,b_i} and the standard deviation σ_{a_i,b_i} of this region are indicated in Equations (2)–(4).

$$\mu_{a_i,b_i} = \frac{\sum_{i=a_i}^{b_i} i * t(l)}{\sum_{i=a_i}^{b_i} t(l)}, \tag{2}$$

$$\sigma^2_{a_i,b_i} = \frac{\sum_{i=a_i}^{b_i} \left(i - \mu_{a_i,b_i}\right)^2}{\sum_{i=a_i}^{b_i} t(l)}, \tag{3}$$

$$\sigma_{a_i,b_i} = \sqrt[2]{\sigma^2_{a_i,b_i}}. \tag{4}$$

Then, Equations (5) and (6) are used to obtain two new thresholds, $THRESHOLD_1$ *and* $THRESHOLD_2$, from the mean μ_{a_i,b_i} and the standard deviation σ_{a_i,b_i} above.

$$THRESHOLD_1 = \mu_{a_i,b_i} - \sigma_{a_i,b_i}, \tag{5}$$

$$THRESHOLD_2 = \mu_{a_i,b_i} + \sigma_{a_i,b_i}. \tag{6}$$

With the two luminance level thresholds $THRESHOLD_1$ and $THRESHOLD_2$, there are three new regions. These three regions are in ranges $[a_i, THRESHOLD_1]$, $[THRESHOLD_1 + 1, THRESHOLD_2 - 1]$, and $[THRESHOLD_2, b_i]$, respectively. In the region $[a_i, THRESHOLD_1]$ and the region $[THRESHOLD_2, b_i]$, two new luminance level thresholds, $THRESHOLD_{a_i,THRESHOLD_1}$ and $threshold_{THRESHOLD_2,b_i}$, are given by Equation (2). The range of the L component in the region $[THRESHOLD_1 + 1, THRESHOLD_2 - 1]$ is the new range. $THRESHOLD_1 + 1$ and $THRESHOLD_2 - 1$ are regarded as a_{i+1} and b_{i+1}. Then, the next two new luminance level thresholds can be computed by Equations (2)–(6).

If the two luminance level thresholds are too close, they cannot effectively separate the images. The following is the limit on the selection of the final luminance level thresholds. The differences between thresholds are calculated and sorted from small to large. If the threshold difference between the previous threshold and the next threshold is less than $(255/(N + 1))$, then the threshold will be deleted. $N \in \{2, 3, \ldots\}$ represents the number of thresholds. After adjusting the number of thresholds, if the total number of remaining thresholds is less than $(N + 1)/2$, then the number of thresholds will be $(N - 1)$.

However, the total number of regions in the target image are used as an index to control the total number of regions in the content image. For example, if the total regions of the target image are less than the total regions of the content image, the proposed method is based on the total regions of the target image and then reduces the number of thresholds of the content image until the thresholds of the two images become the same. Figure 5 shows the results of the adaptive multilevel color transfer between images.

3.1.2. Adaptive Multilevel Color Transfer

Color transfer mainly recolors the content image to the color of the target image. In this paper, the color transfer method proposed by Reinhard et al. [5] is used because this statistically based color transfer method is simpler and faster than other methods. According to the threshold setting in the previous step, each region of the image is color-transferred separately. For example, the regions in Figure 5c,g are color-transferred together.

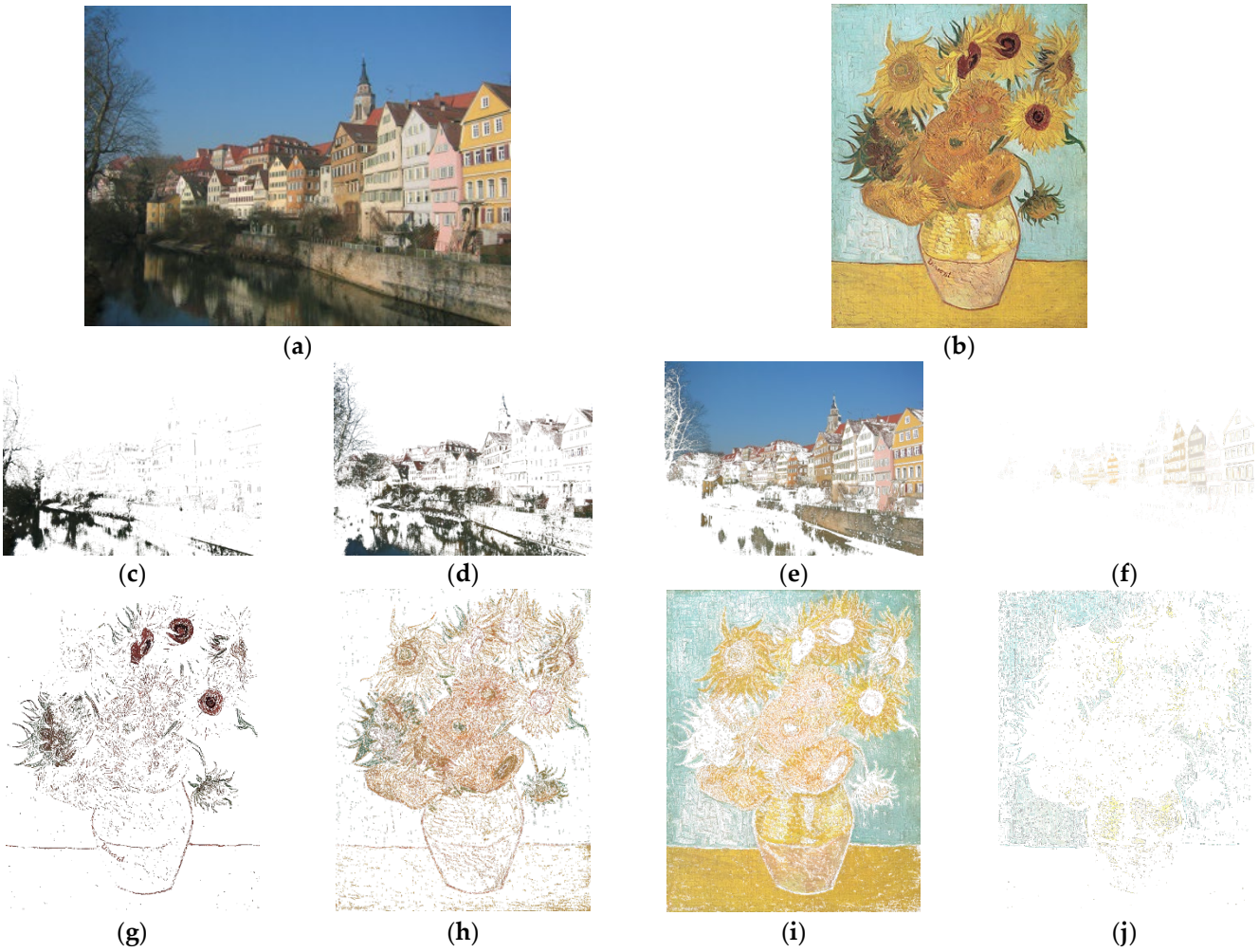


Figure 5. The result of an adaptive multilevel image color transfer between the content image and target image. (**c**–**f**) Three thresholds are set according to the luminance distribution of the content image, and the result of each region in the image is segmented. (**g**–**j**) Three thresholds are set according to the luminance distribution of the target image, and the result of each region in the image is segmented. (**a**) Content image. (**b**) Target image.

Suppose the target image is set to the total t thresholds $\{threshold_{T1}, threshold_{T2}, \ldots, threshold_{Tt}\}$ by luminance distribution, and the content image is also set to the same number of thresholds $\{threshold_{C1}, threshold_{C2}, \ldots, threshold_{Ct}\}$. Color transfer for each pixel of each region $region_r$ for the two images is performed. T and C represent the target image and the content image, respectively.

$$region_r, r \in \{1, 2, 3 \ldots\}. \tag{7}$$

For example, the pixel value in the region $region_{T1}$=$[0 - threshold_{T0}]$ of the target image color-transfers with the pixel value in the same region $region_{C1} = [0 - threshold_{C0}]$ of the content image. Then, the regions of the two images are displayed as $region_{T2} = [threshold_{T0,} threshold_{T1}]$ and $region_{C2} = [threshold_{C0,} threshold_{C1}]$. The process continues until all the regions have finished the color transfer. Equations (8) and (9) calculate the mean of each region of the target image $\mu_{region_{Tr}}$ and the mean of each region of the content image $\mu_{region_{Cr}}$. H and W represent the length and width of the image, and S denotes the total number of region pixels.

$$\mu^{r,g,b}_{region_{Tr}} = \frac{1}{S^{r,g,b}_{region_{Tr}}} T^{r,g,b}_{region_{Tr}}(x, y), \tag{8}$$

$$\mu^{r,g,b}_{region_{Cr}} = \frac{1}{S^{r,g,b}_{region_{Cr}}} C^{r,g,b}_{region_{Cr}}(x,y). \tag{9}$$

Equations (10) and (11) calculate the standard deviation of each region of the target image $\sigma_{region_{Tr}}$ and that of the content image $\sigma_{region_{Cr}}$ as follows:

$$\sigma^{r,g,b}_{region_{Tr}} = \sqrt{\frac{1}{S^{r,g,b}_{region_{Tr}}} \left(T^{r,g,b}_{region_{Tr}}(x,y) - \mu^{r,g,b}_{region_{Tr}} \right)^2}, \tag{10}$$

$$\sigma^{r,g,b}_{region_{Cr}} = \sqrt{\frac{1}{S^{r,g,b}_{region_{Cr}}} \left(C^{r,g,b}_{region_{Cr}}(x,y) - \mu^{r,g,b}_{region_{Cr}} \right)^2} \tag{11}$$

where $\{r,\ g,\ b\}$ stands for a specific channel in the RGB color space. The mean and the standard deviations are calculated separately for these three channels.

The region of each content image corresponds to the region of the target image, and the output is the color-transferred content image $O_{region_r}(x,y)$ using Equation (12).

$$O_{region_r}(x,y) = \frac{\sigma^{r,g,b}_{region_{Tr}}}{\sigma^{r,g,b}_{region_{Cr}}} \left(C^{r,g,b}_{region_{Cr}}(x,y) - \mu^{r,g,b}_{region_{Cr}} \right) + \mu^{r,g,b}_{region_{Tr}}. \tag{12}$$

Each region is overlapped, and the final output image is $O_{color\ transfer}$ using Equation (13) as follows:

$$O_{color\ transfer} = O_{region_1} + O_{region_2} + \cdots + O_{region_n}. \tag{13}$$

That is, the resulting image is processed by the adaptive multilevel color transfer with the content image and the target image. Figure 6 shows the result of the adaptive multilevel color transfer.

3.2. Style Transfer Phase

In this section, judging the effective features in each layer of convolutional layer is introduced in the first phase. The second phase demonstrates style transfers between images.

3.2.1. Feature Visualization and Choose Layers

In recent years, deep learning has been used to extract the features of images. This paper adopted the first 16 layers of VGG19 and removed the last three fully connected layers. Since the convolutional kernel of VGG 19 is relatively small, the required number of parameters is also small. In addition, the learning feature of the multilayered Conv+ReLU is better than that of the single layer Conv+ReLU. The target image is viewed as an input image; through the VGG19 model, the feature extracted by each convolutional layer can be seen. According to Figure 7, VGG19 has five main layers shown in red, orange, yellow, green, and blue blocks. Each color block represents a major layer. There are two or four sublayers for each major layer. Each of the first two major layers (red block and orange block) has two sublayers. Each of the last three layers (yellow block, green block and blue block) has four sublayers. For example, in the red block, this major layer contains two sublayers: *conv1_1* and *conv1_2*.

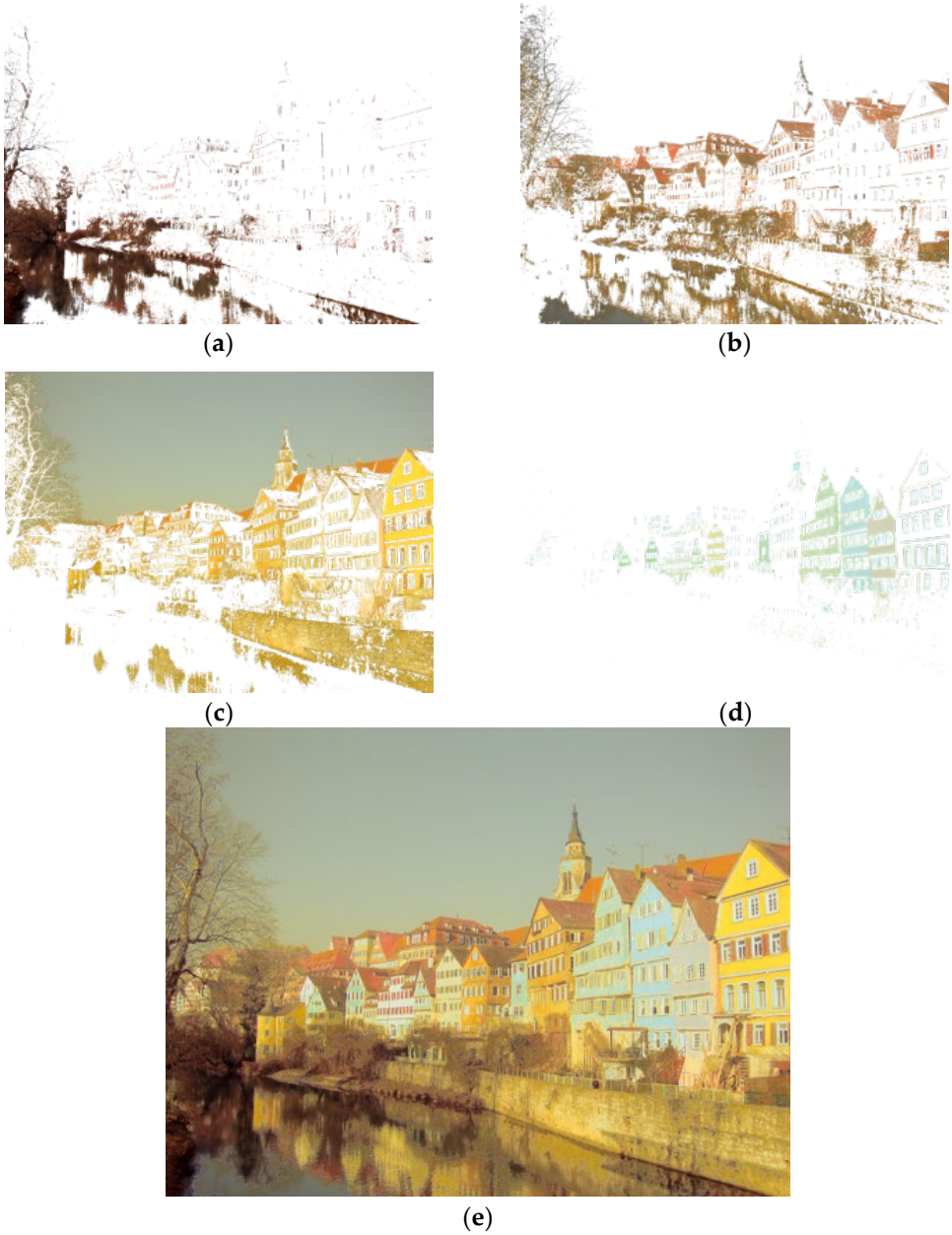


Figure 6. The result of the adaptive multilevel image color transfer. (**a**–**d**) Color transfer with each region. (**e**) The result of overlapping the image (**a**–**d**).

In the feature map, although many green blocks represent zero, the green grid still contains some yellow parts. Here, the effective feature map of each layer is calculated to select a layer for the image. All sublayers of each major layer are compared together. The first major layer—that is, the red block—is calculated to select one of the two sublayers. The first and second major layers mainly use the structural similarity index (SSIM index), the method proposed by Wang et al. [25], to calculate the effective feature extraction in each sublayer. The structural similarity index (SSIM index) is an indicator used to measure the similarity of two images. SSIM compares the images based on their structure, luminance, and contrast.

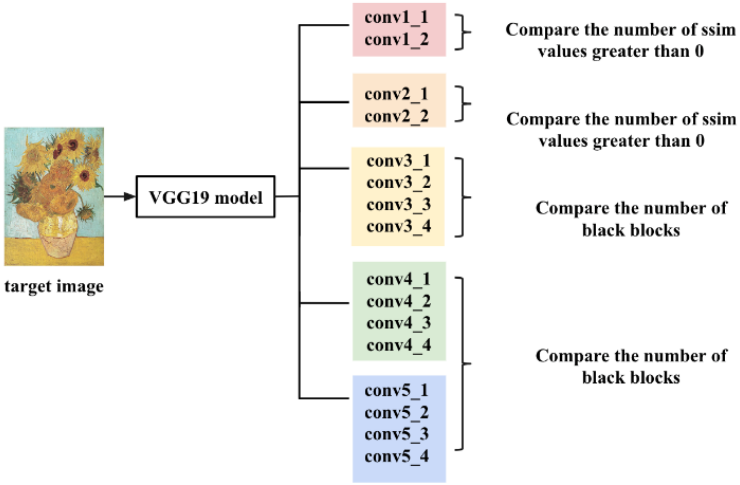


Figure 7. Architecture of the number of layers selected for the target image.

When an image is identified by a filter, a black block condition may occur. The black block indicates that the filter does not recognize valid features. Therefore, the third, fourth, and fifth major layers mainly use the number of black blocks in the feature map to calculate the effective feature extraction in each sublayer. Figure 7 displays the architecture of the selection of a convolutional layer for the target image.

Next, Figure 7 also demonstrates how the first two major layers use SSIM to see the degree of extraction of the effective feature map in the convolutional layer. The proposed method comprises an analysis of the features extracted from the target image for each convolutional layer. Using *conv1_1* as an example, after running the 64 filters, there will be 64 feature maps. First, all the feature maps of each sublayer are merged into a new feature map, and then the SSIM values between the feature map in the sublayer x and the new feature map y are calculated. SSIM is shown in Equation (14). All feature maps of this layer are fused in 1:1 size. A new feature map representing the entire layer can be generated since SSIM will calculate the luminance, contrast, and structure of the two images. If the feature is included in the new feature map, it represents the structure of the image that can be captured.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{14}$$

where μ_x and μ_y represent the means of images x (the feature map in the sublayer) and y (the new feature map), respectively. σ_x and σ_y represent the standard deviations of images x and y, respectively. σ_{xy} represents the covariance of images x and y. C_1 and C_2 are constants that maintain the stability of the formula. The scope of SSIM is $[-1,\ 1]$. The larger the value of SSIM, the more similar the images. If the value of SSIM is greater than 0, it is judged to be a valid feature. The number of valid features can be counted within the sublayer. Taking the first major layer as an example, the effective feature numbers of the two sublayers are respectively calculated and compared. A convolutional layer with a larger number of effective features is judged to be the most capable feature map in this major layer. Figure 8 displays a process describing how to use SSIM to determine the effective features in a convolutional layer of the sunflower image.

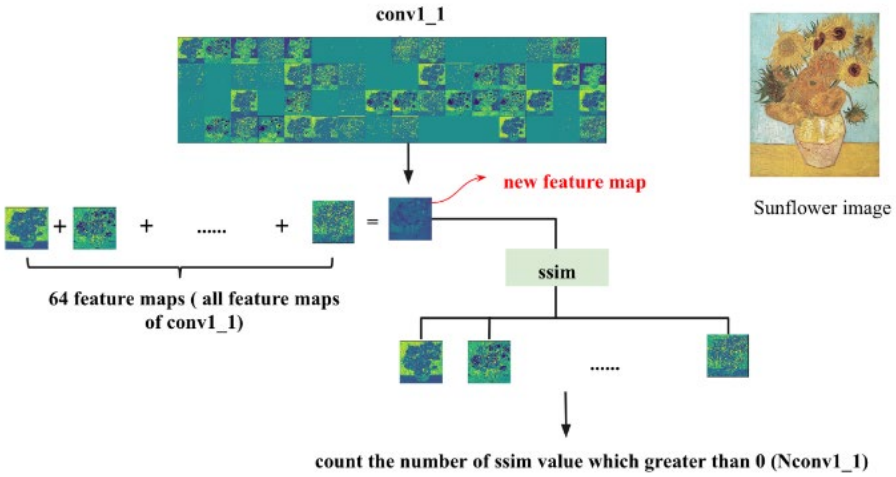


Figure 8. The process of comparing each feature map through SSIM.

The last three major layers use the number of black blocks to choose layers. In this part, the total number of black blocks is counted in each layer. The convolutional layer with the smallest number of total black blocks is selected. This paper compared the sublayers of the third major layer and chose a convolutional layer with a small number of black blocks. Similarly, since the fourth and fifth major layers have the same total number of filters, only one sublayer with the least black blocks is selected from the two major layers. In addition, the fourth and fifth major convolutional layers have the same number of filters, so they are judged together as well. The diversity of features increases as the convolutional layers are selected based on different major convolutional layers.

3.2.2. Image Style Transfer

This step is based on the deep learning style transfer method. The VGG19-based method of Gatys et al. was applied in this paper. The lower-level convolutional layer captures color patches, simple lines, or colors, whereas the higher-level convolutional layer captures the entire object of the image, such as the image architecture. Therefore, a convolutional layer is selected for the content image, $conv4\_2$. The loss function $\mathcal{L}_{color\ transfer}$ between the color-transferred content image $O_{color\ transfer}\ (C)$ obtained from the color transfer step and the generated image $(Generated)$ is received. The loss function $\mathcal{L}_{target}$ between the target image (T) and the generated image is calculated based on the selected number of specific convolutional layers. In Equation (15), the ratio of $\alpha/\ \beta$ is set at 1/1000. Different values of α and β are employed to get the final loss function $\mathcal{L}_{generated}$. Figure 9 shows the image style transfer process.

$$\mathcal{L}_{generated} = \alpha\mathcal{L}_{colortransfer}(conv4_2) + \beta\mathcal{L}_{target} \quad \text{(Specific number of layers).} \tag{15}$$

Next, the calculations of the loss function $\mathcal{L}_{color\ transfer}$ and $\mathcal{L}_{target}$ are introduced in the following. 1. Between the color-transferred content image (C) and generated image $(Generated)$: the loss function $\mathcal{L}_{colortransfer}$ between the color-transferred content image (C) and the generated image $(Generated)$ is represented in Equation (16) below:

$$\mathcal{L}_{colortransfer} = \frac{1}{2} \parallel c^l - g^l \parallel^2, \tag{16}$$

where c^l and g^l respectively indicate the color-transferred content image and the generated image corresponding to each other in the convolutional layer l. The error squared loss function between the two values is calculated. The smaller the loss function $\mathcal{L}_{colortransfer}$, the more similar the color-transferred content image and the target image at the same location. 2. Between the target image (T) and generated image $(Generated)$: in the same convolu-

tional layer, any two features are made using the Gram matrix G^l_{ij} as the inner product to find the correlations between the features. These correlations can represent information about some of the styles. That is to say, after finding multiple feature correlations between multiple convolutional layers, the style of the entire image can be obtained. Considering the target image and the generated image in the VGG19 model, the Gram matrix of the convolutional layer l is represented by Equations (17) and (18). $a(T)^l_{c1}$ and $a(T)^l_{c2}$ respectively represent any two activation values of the target image. $a(G)^l_{c1}$ and $a(G)^l_{c2}$ respectively represent any two activation values of the generated image.

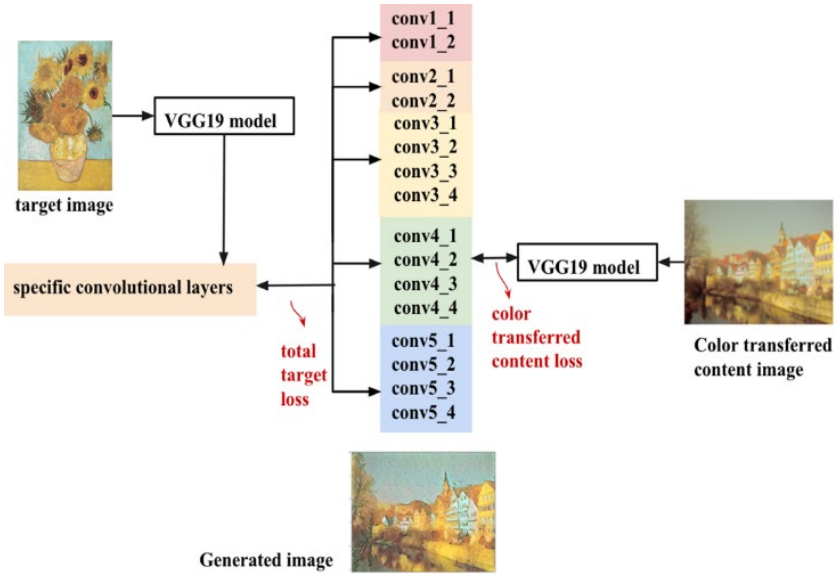


Figure 9. The image style transfer process.

$$T^l_{c1,c2} = \sum_{width} \sum_{height} \sum_{length} a(T)^l_{c1} a(T)^l_{c2}, \tag{17}$$

$$G^l_{c1,c2} = \sum_{width} \sum_{height} \sum_{length} a(G)^l_{c1} a(G)^l_{c2}, \tag{18}$$

n_{length}, n_{width}, and n_{height} represent the length, width, and height of the convolution layer l. T^l indicates the feature that the target images correspond to each other in the convolutional layer. The error squared loss function $\mathcal{L}_{target}$ of the target image and the generated image is calculated, and the weight of each layer can be set as ω_l, both of which are displayed in Equation (19).

$$\mathcal{L}_{target} = \sum_{l=0}^{L} \omega_l \left(\frac{1}{\left(2n^l_{width} n^l_{length} n^l_{height}\right)^2} ||G^l - T^l||^2 \right), \tag{19}$$

4. Experimental Results

In this section, Section 4.1 introduces the experimental image setting and evaluation method. Next, Section 4.2 presents the experimental results of the proposed method. Finally, the experimental results of the proposed method are compared with related works on color transfer and related works on style transfer in Section 4.3, respectively.

4.1. Performance Evaluation

In the experiment, the input images contain the content image and the target image in a size of 512 × 512. The color transfer result of the image is evaluated by the histogram of the image and measured by the distance between the two image histograms. The histogram distance here is measured by Euclidean geometry. The image color histogram similarity is

evaluated in Equation (20). The greater the similarity, the more similar the color distribution of the representative image.

$$Similarity(h(C), h(T)) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{|c_i - t_i|}{Max(c_i, t_i)}\right), \tag{20}$$

where $h(C)$ and $h(T)$ are histograms of the color transferred image and the target image, respectively. N represents the range of the histogram. The larger the value of the similarity, the more similar the two images. If the two images are completely similar, they will be indicated as 1. Conversely, if they are completely dissimilar, the similarity will be expressed as 0.

The style transfer result of the image is evaluated by Equations (21)–(23). The comparison uses the variance of the images. Each image generates a variance of the pixels, and this variance represents the feature value of the image. Therefore, the proximity of the variance of the two images is compared to determine the similarity between the two images.

$$Mean_l = \frac{\sum_{j=1}^{w} p(i,j)}{w} i = (1, 2, \ldots, l), \tag{21}$$

$$Mean_{image} = \frac{\sum_{i=1}^{l} Mean_l}{l}, \tag{22}$$

$$Variance = \frac{\sum_{i=1}^{l} (Mean_l - Mean_{image})^2}{l}. \tag{23}$$

The image size can be flexibly scaled according to the user's needs. Since the similarity and color correlation of the image are not large, the image is converted into a gray image to reduce the computational complexity. l and w represent the length and width of the image, respectively. $Mean_l$ of each row of pixels and $Mean_{image}$ of the pixels of the whole image are calculated. Each average $\mathrm{Mean_l}$ corresponds to a feature of a row. The standard deviation between the average of each row and the average of the whole image is calculated, and then the standard deviations are all added up. Variance is calculated for all averages, and this variance is the characteristic value of the image. The similarity of the image is determined according to the difference between the variances of the two images.

The smaller the difference between the variances of the two images, the more similar the two images. If the value of the difference between the variances of the two images is 0, then the features of the two images are completely similar.

4.2. Experimental Results of the Proposed Method

According to the proposed method in this paper, the content image and target image were converted into the CIELAB color space. The threshold for the L channel of the two images was set. The thresholds for the target image were set to 86, 139, 207, and 228. According to the limit of threshold in this proposed method, the final thresholds of the target image were set at 86, 139, and 207. The total number of thresholds must be adjusted according to the target image. The thresholds for the content image were set to 46, 86, 167, and 206. It could be seen that the difference between the thresholds of the content image was less than 255/5 regions. However, in order to reach the same threshold number of the target image, the proposed scheme would delete the threshold with the smallest difference. Given the threshold limit, the final thresholds of the content image were set to 46, 86, and 206. Then, the color transfer was performed in the same region of the two images. For example, the region between 0 and 86 in the target image underwent the color transfer with the region between 0 and 46 in the content image.

After the image color transfer was performed, SSIM and the total number of black blocks were used to select valid features for the convolutional layer. The first major layer and the second major layer were selected by SSIM, where *conv1_2* and *conv2_2* were selected. After the third layer, the fourth layer, and the fifth layer combined the sublayer features,

the sublayer fusion maps in the major layer were not much different. As a result, there was no way to find the effective features with SSIM. Thus, we selected *conv3_4* and *conv5_2* according to the number of black blocks. Figure 10 shows the black blocks situation and uses *conv3_1* as an example. Therefore, we could only make judgments based on the black blocks.

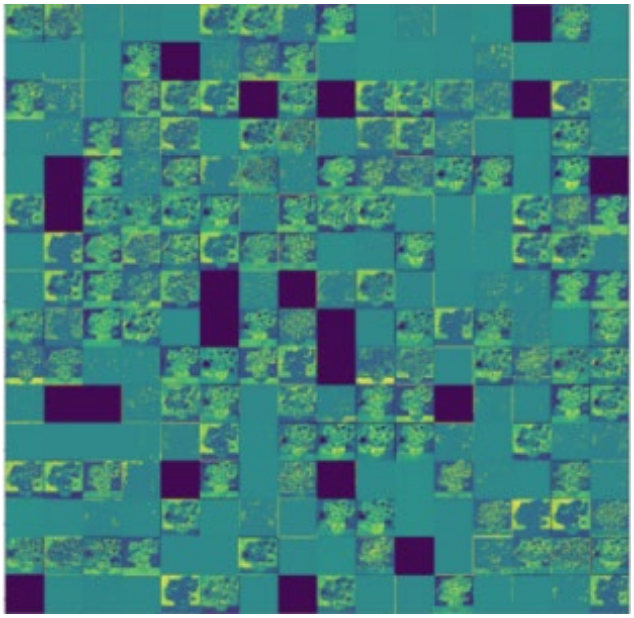


Figure 10. Visualization of the feature map of the convolutional layer *conv3_1*.

4.3. Image Analysis

Figure 11 shows the fusion image of all the feature maps in each convolutional layer. Figure 11a,b shows the fusion feature maps of the sublayer in the first major convolutional layer. Figure 11c,d shows the fusion feature maps of the sublayer in the second major convolutional layer. Figure 11e–h is the fusion feature maps of the third major convolutional layer. Figure 11i–l is the fusion feature maps of the fourth major convolutional layer. Figure 11m–p is the fusion feature maps of the fifth major convolutional layer.

In Figure 12, two images are used as an example of the image color transfer and style transfer. Figure 12c displays the color-transferred image with the adaptive multilevel color transfer, and Figure 12d shows the style-transferred image following Figure 12c. Style losses are computed in conv1_2, conv2_2, conv3_4, conv5_3. Using the impressionist image as the target image, the image transfer results show that the structure, brightness, and darkness of the content image can be preserved. At the same time, the color of the target image is also retained.

In Figure 13, an abstract target image is used to transform the style of the image. It can be seen that the resulting image not only maintains the structure of the dog in the content image, but also can be combined with the texture of the target image. In addition, Figure 14 shows the results of the proposed method with different types of the images.

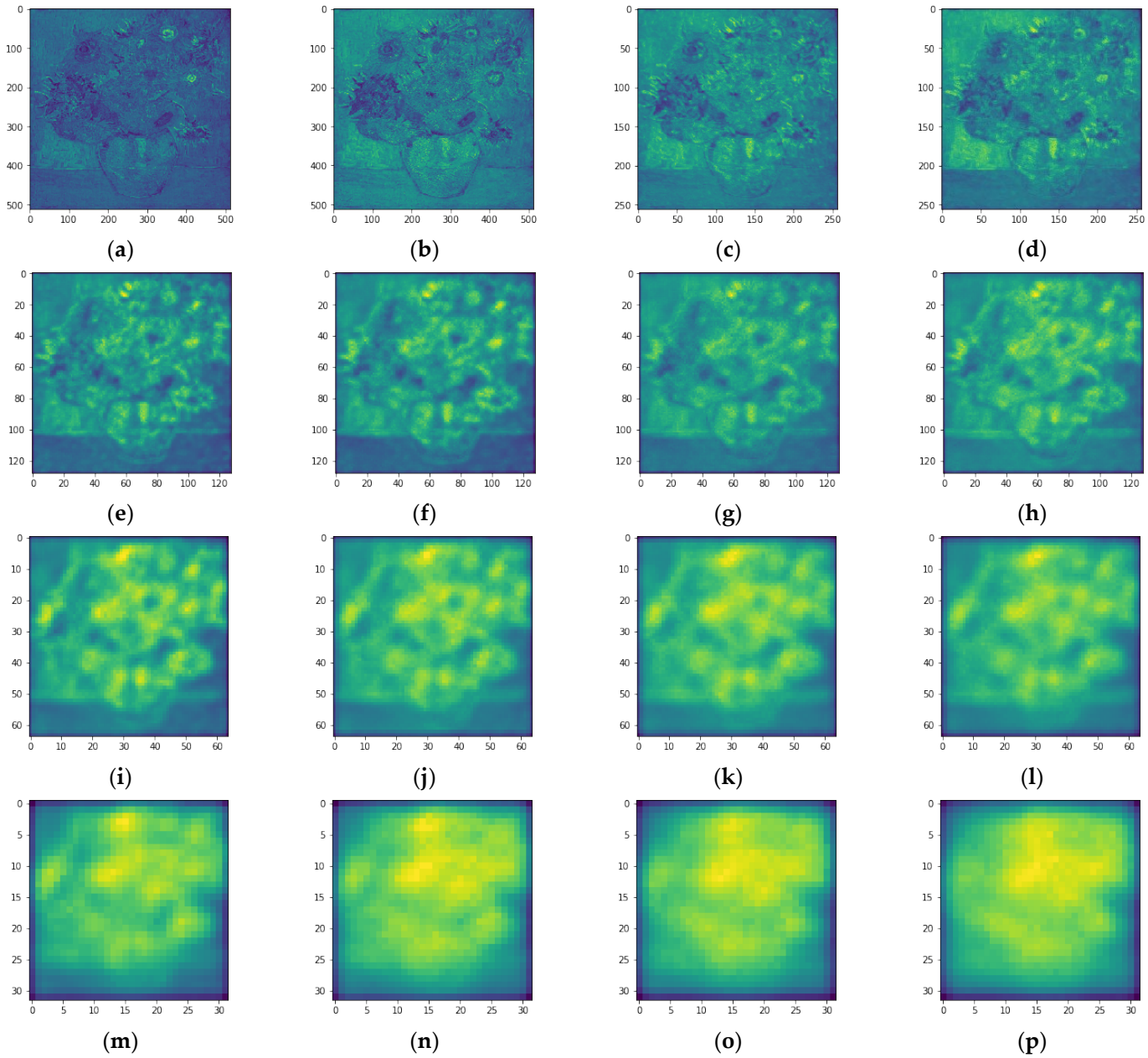


Figure 11. The fusion feature maps of each convolutional layer. (**a**,**b**) shows the fusion feature maps in the first major convolutional layer. (**c**,**d**) shows the fusion feature maps in the second major convolutional layer. (**e**–**h**) is the fusion feature maps of the third major convolutional layer. (**i**–**l**) is the fusion feature maps of the fourth major convolutional layer. (**m**–**p**) is the fusion feature maps of the fifth major convolutional layer.

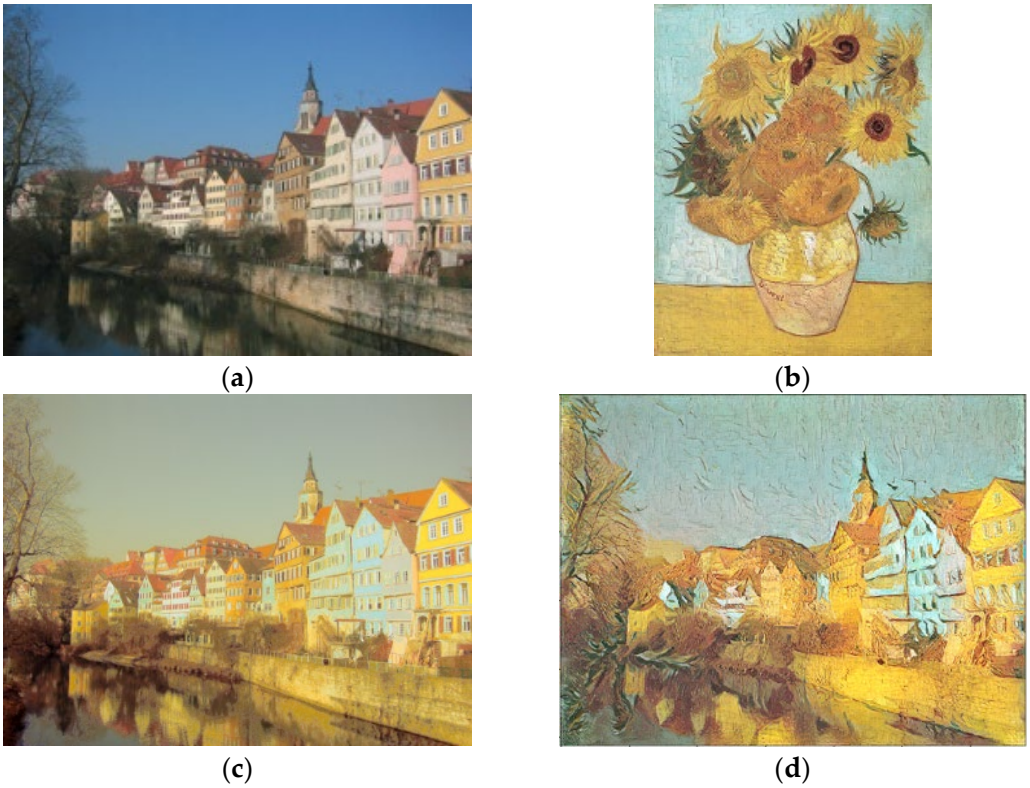


Figure 12. The results of the proposed method image transfer between the landscape image and impressionist image. (**a**) Content image. (**b**) Target image. (**c**) Our color-transferred image. (**d**) Our style-transferred image.

Examples of results using the proposed method and the related works are shown in Figure 15. Figure 15 demonstrates that our results can express the brightness of the content image. The obvious difference is seen from the lower left corner. The results of this proposed method highlight the difference between the lake image and the building in the content image, which is more visually stereoscopic. The result of Reinhard et al.'s method, though visually soft, does not represent the context of the image. The result of Jimenez-Arredondo et al.'s method faintly shows the foreground and background of the image, but the mottled condition appears in the lower left. The visual effect is not smooth. Figure 16 is a comparison of the similarity results of the colors on two images using the pixel histogram to calculate the image similarity between the color-transferred images in Figure 15c–e and the target image in Figure 15b. Although the color similarity of the Jimenez-Arredondo et al.'s method is relatively high, the visual appearance is rather unharmonious. Figure 17 shows the comparison of the image similarity between the content image in Figure 15a and the color transfer images in Figure 15c–e. As displayed in the following Figures 16 and 17, the proposed method retains a more complete structure of the content image. Through the color and structure analysis, our color-transferred image provides a better result.

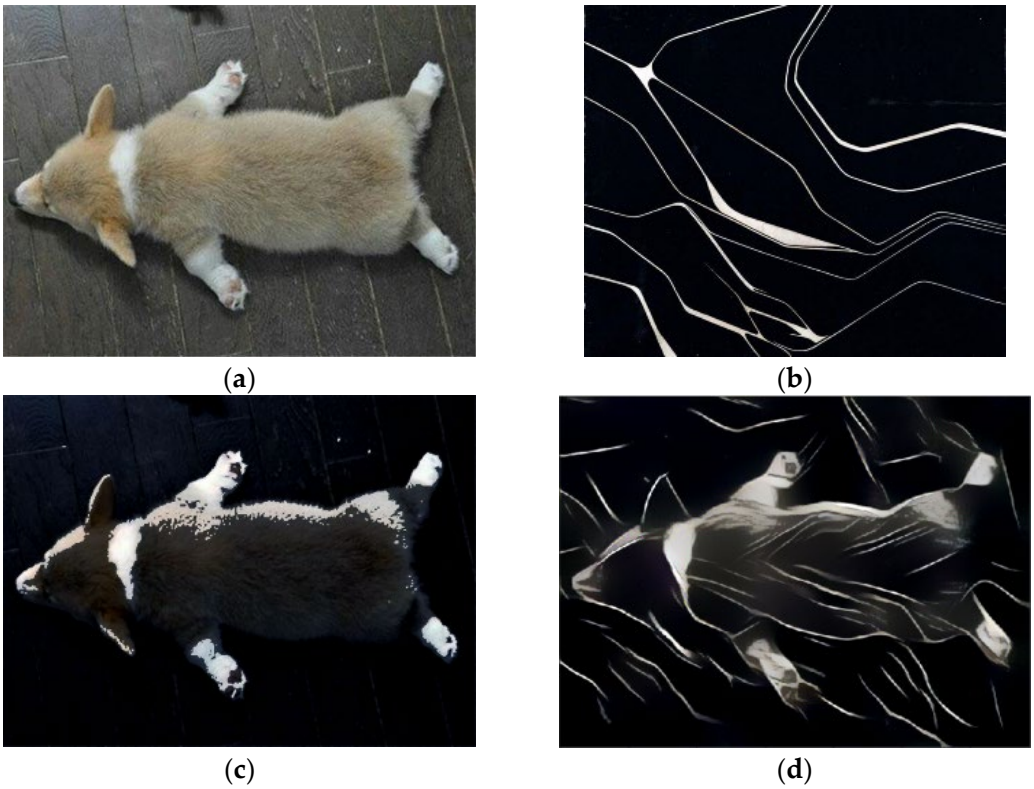
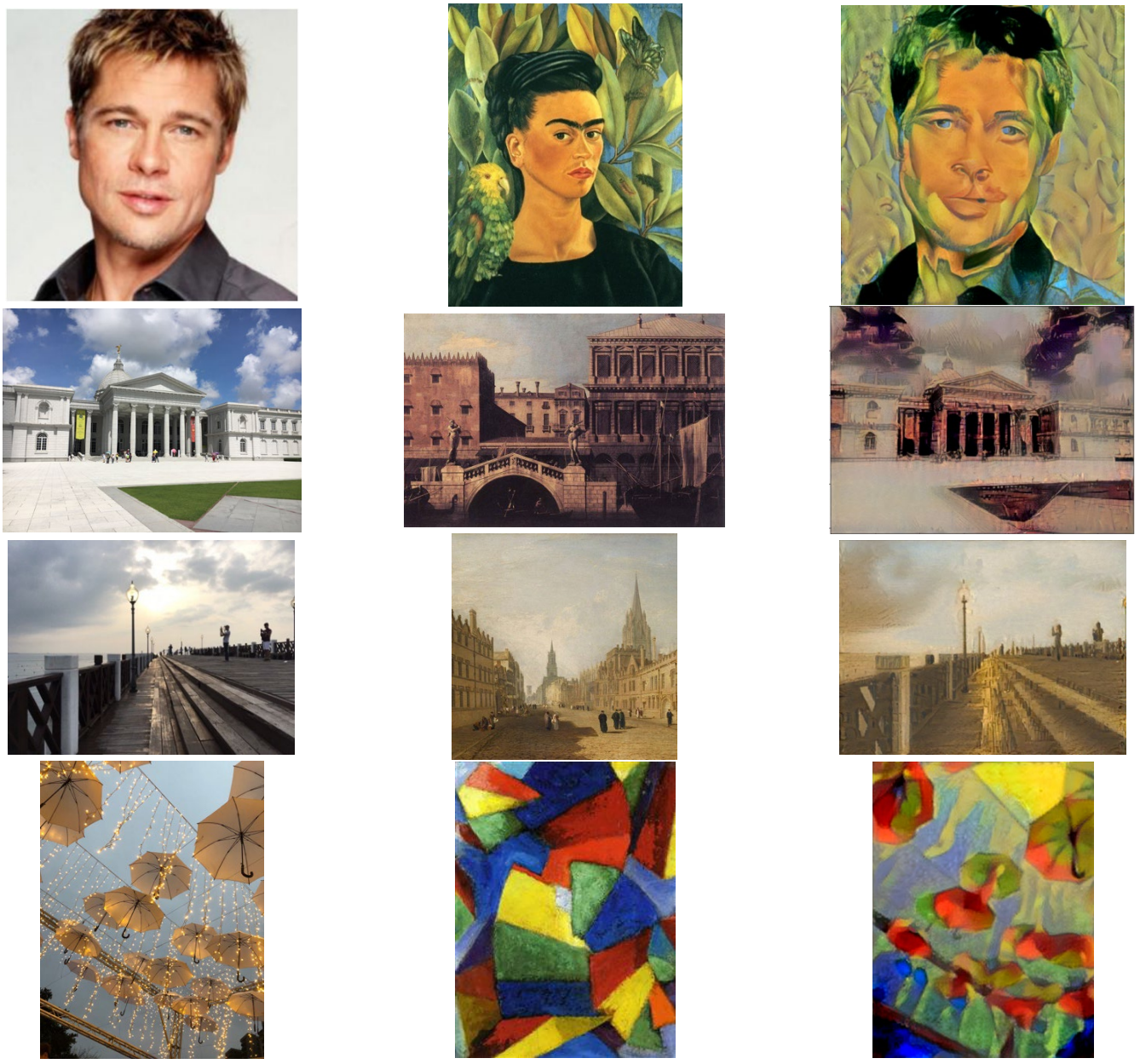


Figure 13. The results of the proposed method image transfer between a wildlife image and an abstract image. (**a**) Content image. From the MS-COCO dataset [26]. (**b**) Target image. From the WikiArt database [27]. (**c**) Color-transferred image with adaptive multilevel image color transfer. (**d**) Style-transferred image after (**c**). Style losses computed on *conv1_2*, *conv2_2*, *conv3_2*, *conv4_1*.

In this paper, SSIM and black blocks were used to describe the effective features by feature visualization. In the neural network, some of the effective features are used to extract texture and details, and some are used to extract contour or shape features. The feature map obtained from each convolutional layer is the output of the activation function ReLU. If this feature strength is weak, it will be represented as 0. The black block represents that most of the values are zero after passing the activation function ReLU, and the filter cannot recognize the feature. When there are fewer effective features, the result of image style conversion may be affected.

In Figure 18, the proposed image style transfer method demonstrates better results in detail than the method of Gatys et al. For example, the windows on the building are more obvious, and the structure of the content image is more consistent. The color on the lake is also in line with the target image. Figure 19 compares the resulting image with the variance of the target image. The closer the variance of the two images, the more similar they are. Therefore, it can be seen from the data that the image transfer result of this paper was closer to the target image. At the same time, the proposed scheme also scaled the image to two different lengths to compare their similarities.



(**a**) (**b**) (**c**)

Figure 14. The results of the proposed method image transfer with different types of images. (**a**) Content image. (Top: "Brad Pitt" [28].) (**b**) Target image. From the WikiArt database. (**c**) After completing the adaptive multilevel image color transfer and selecting specific convolution layers according to the target image, the final result of the style transfer is displayed.

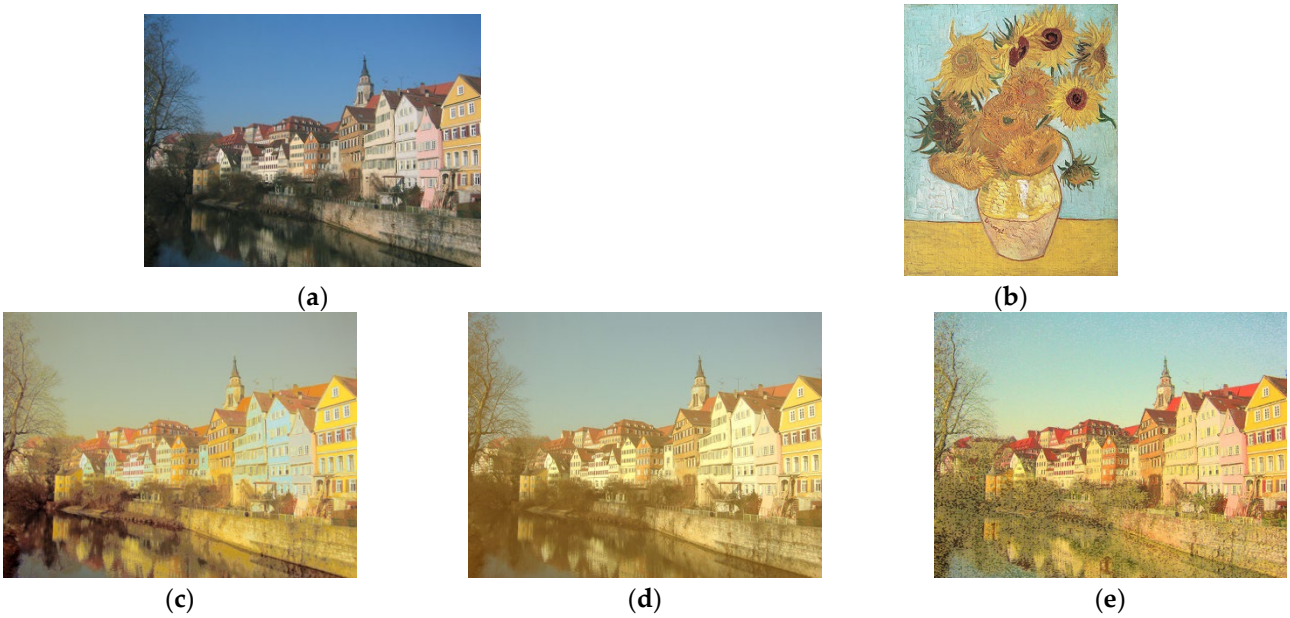


Figure 15. Comparison with Reinhard et al.'s method and Jimenez-Arredondo et al.'s method. (**a**) Content image. (**b**) Target image. (**c**) The result image of our color-transferred method. (**d**) The result of Reinhard et al.'s method. (**e**) The result of Jimenez-Arredondo et al.'s method.

Target image	Reinhard *et al.*'s method (2001)	Jimenez-Arrdondo *et al.*'s method (2017)	Our color transferred image
			
Image color similarity	**0.401**	**0.593**	**0.489**

Figure 16. Image color similarity according to the histogram of the target image, other related color transfer resulting images, and our color-transferred image.

Content image	Reinhard *et al.*'s method (2001)	Jimenez-Arrdondo *et al.*'s method (2017)	Our color transferred image
			
Image similarity	**189.242**	**180.586**	**15.748**

Figure 17. Image similarity between the content image, other related color transfer resulting images, and our color-transferred image.

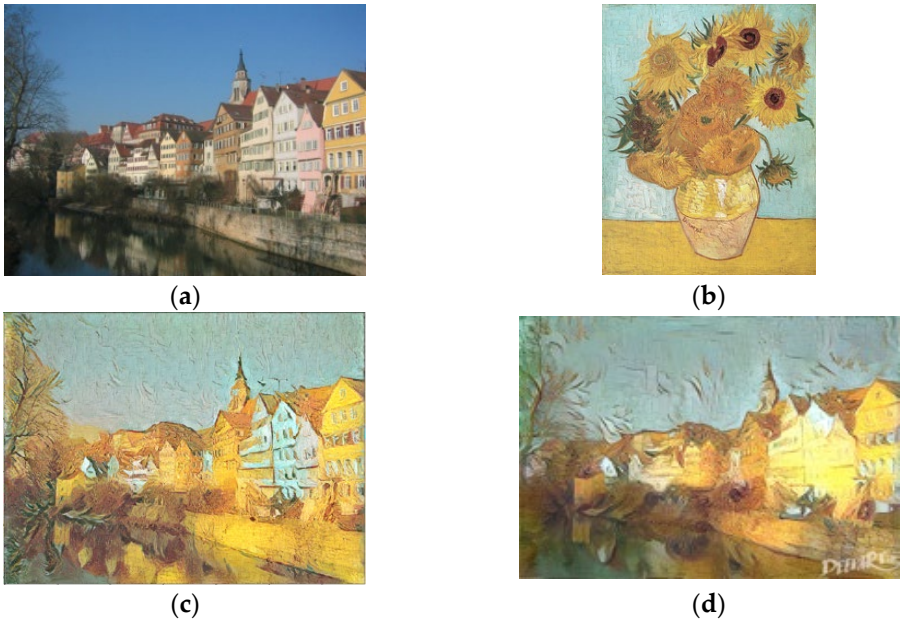


Figure 18. Comparison with Gatys et al.'s method. (**a**) Content image. (**b**) Target image. (**c**) Our results. (**d**) The result of Gatys et al.'s method, with the content features on layer conv4_2 and the style features on layer conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1.

Target image	Gatys *et al.*'s method (2016) (Result from Deepart.io (2019))	Our style transfer image
		
Image similarity with length=512	**371.985**	**192.660**
Image similarity with length=64	**356.996**	**146.588**

Figure 19. Image similarity with the variance of the images.

Figure 20 displays the comparison of the results of our proposed method, Gatys et al.'s method, and X. Li et al.'s method. It can be seen that the result of Gatys et al.'s method is not obvious, but it can retain the style of target image. On the contrary, the result of X. Li et al.'s method has only a few styles, but the contour of the dog is obviously prominent. As to the result of our method, our image maintains a balance between the structure of

the content image and the style of the target image, indicating that it can distinguish the structure of the dog as well as preserve the style of other images.

In Figure 21, there is another example comparing the result of our method with the results of Gatys et al.'s method and X. Li et al. 's method. The three methods are used to convert the portrait image and the hand-drawn portrait image. From the results of the three different methods, it is apparent that the result of the proposed method more accurately expressed the contour of the content image.

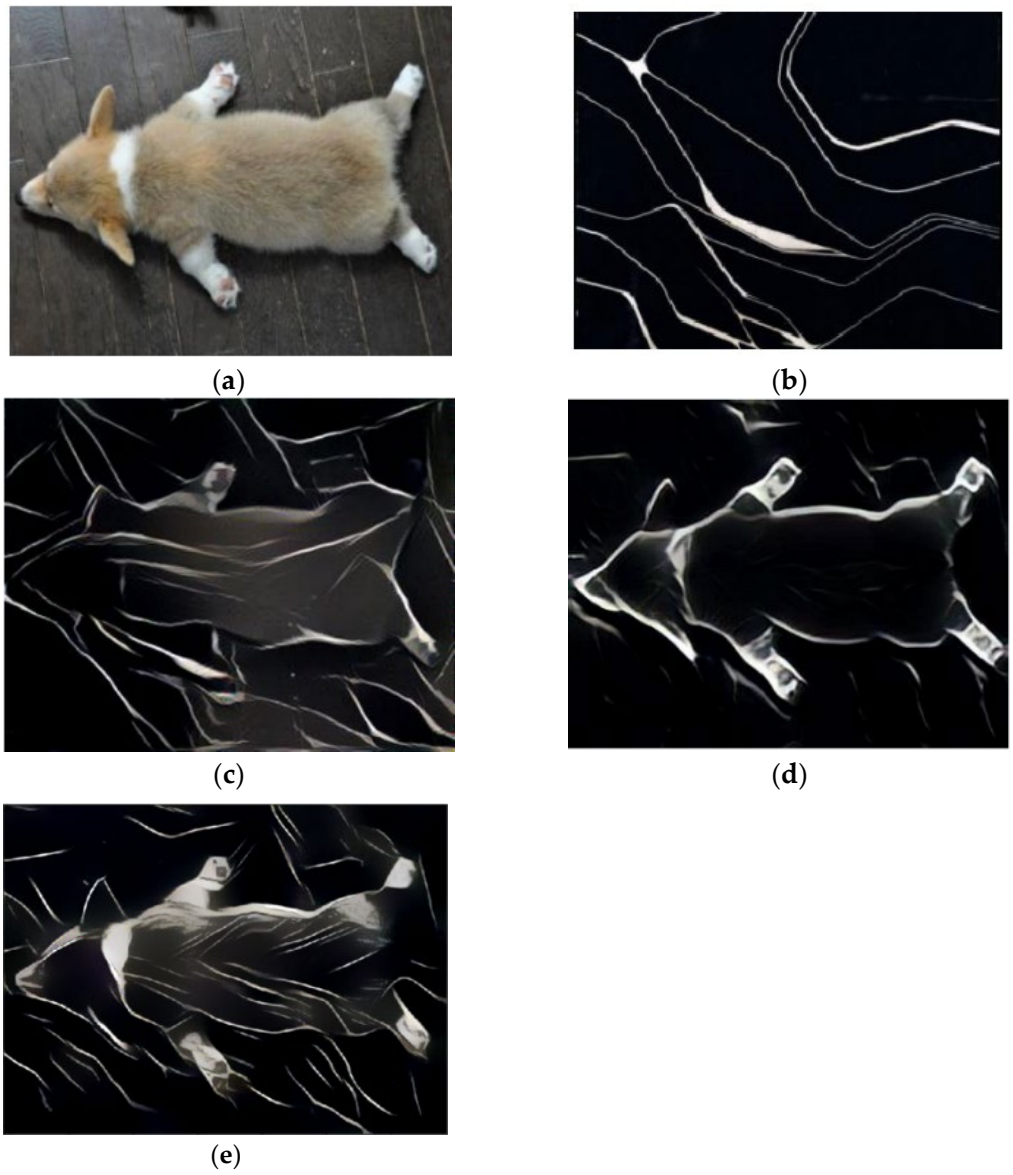


Figure 20. The comparison of Gatys et al.'s method, X. Li et al.'s method, and the proposed method. (**a**) Content image. (**b**) Target image. (**c**) The results of Gatys et al.'s method. (**d**) The results of X. Li et al.'s method. (**e**) Our result.

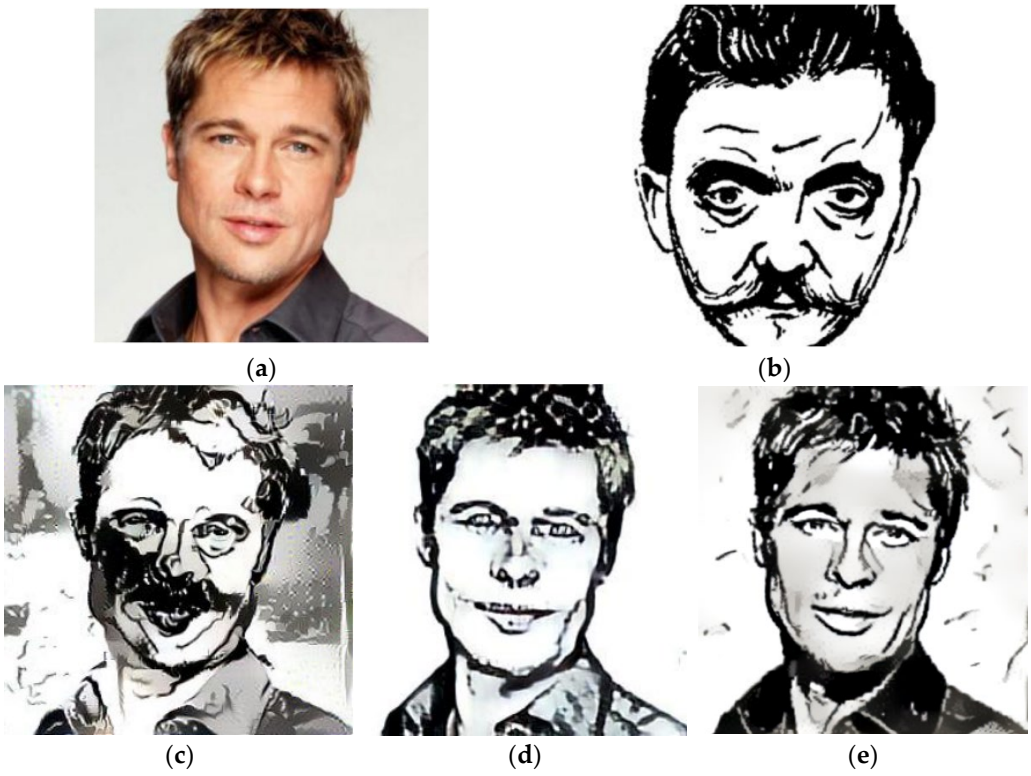


Figure 21. The comparison of the methods of Gatys et al., X. Li et al., and the proposed method. (**a**) Content image. (**b**) Target image. (**c**) The results of Gatys et al.'s method. (**d**) The results of X. Li et al.'s method. (**e**) Our result.

5. Conclusions

The proposed method is a deep learning image transfer technique based on the color transfer method. In this paper, the distribution of luminance was used to segment the image, and then the color transfer was performed for different regions. The image color transfer method proposed in this paper is a local color transfer, so it can improve the results of using global color transfer. To imitate the basic brushstrokes of the artist, this paper proposes a method to select the effective features in the convolutional layer. Using the evaluation method of variance, the color of the image affects the correspondence of the deep learning features. The proposed method is characterized by the color transfer of images using their luminance distribution and then selecting the convolutional layer with effective features. At the same time, good visual results can be obtained without any manual adjustment. By evaluating the similarity between the resulting image and the target image, it was demonstrated that the method in this paper can reduce the difference gap by more than two times. In addition, our image transfer results can show the balance between the color and style of the target image. Thus, the method proposed in this paper can successfully simulate artistic painting in terms of image style transfer technique.

Author Contributions: All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by H.-C.W., Y.-C.L., Y.-Y.C. and Y.-Y.W. The first draft of the manuscript was written by H.-C.W., and all authors commented on previous versions of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Data Availability Statement: The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interests.

References

1. Galanter, P. What is generative art? complexity theory as a context for art theory. In Proceedings of the GA2003-6th Generative Art Conference, Milan, Italy, 10–12 December 2003.
2. Jimenez-Arredondo, V.H.; Cepeda-Negrete, J.; Sanchez-Yanez, R.E. Multilevel color transfer on images for providing an artistic sight of the world. *IEEE Access* **2017**, *5*, 15390–15399. [CrossRef]
3. HaCohen, Y.; Shechtman, E.; Goldman, D.B.; Lischinski, D. Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph.* **2011**, *30*, 1–10. [CrossRef]
4. Levin, A.; Lischinski, D.; Weiss, Y. Colorization using optimization. *ACM Trans. Graph.* **2004**, *23*, 689–694. [CrossRef]
5. Reinhard, E.; Adhikhmin, M.; Gooch, B.; Shirley, P. Color transfer between images. *IEEE Comput. Graph. Appl.* **2001**, *21*, 34–41. [CrossRef]
6. Papadakis, N.; Provenzi, E.; Caselles, V. A variational model for histogram transfer of color images. *IEEE Trans. Image Process* **2011**, *20*, 1682–1695. [CrossRef] [PubMed]
7. Cepeda-Negrete, J.; Sanchez-Yanez, R.E.; Correa-Tome, F.E.; Lizarraga-Morales, R.A. Dark image enhancement using perceptual color transfer. *IEEE Access* **2017**, *6*, 14935–14945. [CrossRef]
8. Liu, S.; Sun, H.; Zhang, X. Selective color transferring via ellipsoid color mixture map. *J. Vis. Commun. Image Represent* **2012**, *23*, 173–181. [CrossRef]
9. Khan, A.; Jiang, L.; Li, W.; Liu, L. Fast color transfer from multiple images. *Appl. Math.-A J. Chin. Univ.* **2017**, *32*, 183–200. [CrossRef]
10. Liu, S.; Pei, M. Texture-aware emotional color transfer between images. *IEEE Access* **2018**, *6*, 31375–31386. [CrossRef]
11. Gatys, L.; Ecker, A.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
12. Li, X.; Liu, S.; Kautz, J.; Yang, M.H. Learning linear transformations for fast arbitrary style transfer. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
13. Olah, C.; Mordvintsev, A.; Schuber, L. Feature visualization. *Distill* **2017**, *2*, e7. [CrossRef]
14. Liu, S.; Zhu, T. Structure-guided arbitrary style transfer for artistic image and video. *IEEE Trans. Multimed.* **2021**, *24*, 1299–1312. [CrossRef]
15. Huang, J.; Liao, J.; Kwong, S. Semantic example guided image-to-image translation. *IEEE Trans. Multimed.* **2021**, *23*, 1654–1665. [CrossRef]
16. Ruderman, D.; Cronin, T.; Chiao, C. Statistics of cone responses to natural images: Implications for visual coding. *J. Opt. Soc.* **1998**, *15*, 2036–2045. [CrossRef]
17. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
19. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 2015 International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
20. Vedaldi, A.; Fulkerson, B.; Lenc, K.; Perrone, D.; Perdoch, M.; Sulc, M.; Sarbortova, H. CNNs for MATLAB. MatConvNet. 2015. Available online: http://www.vlfeat.org/matconvnet/models/beta16/imagenet-vgg-verydeep-19.mat (accessed on 20 February 2019).
21. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009.
22. Fairchild, M. *Color Appearance Models*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
23. Schanda, J. *Colorimetry: Understanding the CIE System*; John Wiley & Sons: Hoboken, NJ, USA, 2007; pp. 25–76.
24. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans.* **1979**, *9*, 62–66. [CrossRef]
25. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process* **2004**, *13*, 600–612. [CrossRef] [PubMed]
26. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollr, P. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (2014 ECCV), Zurich, Switzerland, 6–12 September 2014.
27. Nichol, K. Painter by Numbers. wikiart. 2016. Available online: https://www.kaggle.com/c/painter-by-numbers (accessed on 5 March 2019).
28. Bethge, M.; Ecker, A.; Gatys, L.; Kidzinski, L.; Warchol, M. Turn Your Photos Into Art. DeepArt Retrieved. 2019. Available online: https://deepart.io/ (accessed on 15 January 2019).