# The Research and Implementation of Image Style Conversion Algorithm Based on Deep Convolutional Neural Network

Huang Yaoqun(✉), Xia Hongyang, and Kang Hui

School of Electronics and Information Engineering, Heilongjiang University of Science and Technology, Harbin 150022, China
huangyaoqun@126.com

**Abstract.** With the development of deep learning and image technology, the deep convolutional neural network has been widely used to deal with image problems. In this paper, the pretrained vgg-19 convolutional network model is adopted to extract and define the loss function according to the image characteristics, and preset model parameters. The model training is completed through reverse propagation gradient descent and optimization iteration, finally, the artistic painting style conversion of photos is realized. At the same time, by adjusting the size of style weight and content weight, the output image is more inclined to the style picture, or more inclined to the content picture. Finally, the objective evaluation of the output image is has been completed by comparing the image style conversion results of the TensorFlow and the PyTorch. The results show that under the same iteration times, the PyTorch framework has relatively small computation, fast processing speed, and better image color retention effect, in contrast the TensorFlow frame retains more features of style images.

**Keywords:** Image style conversion · Deep learning · VGG-19 convolutional network model

## 1 Introduction

The image style conversion is a technology that extracts the image content contained in one image and the image style contained in the other model for synthesis, and then formed a new model, which can be widely used in animation production, advertising design, mobile image processing, and other fields.

Before the rise of deep learning, the image style conversion methods used by people were difficult to meet the actual needs, and the synthesized images were relatively rough. With the maturity of computer technology and the fierce development of deep learning, the convolutional neural network was adopted to image style conversion, have changed the concept of image style conversion, make the processing of image style conversion is no longer at the pixel level, still a global translation by extracting image features, in term of tonal, texture and spatial relations of image conversion [1], its basic principle is that

X. Jiang and P. Li (Eds.): GreeNets 2020, LNICST 333, pp. 350–359, 2020.
https://doi.org/10.1007/978-3-030-62483-5_36

by establishing loss function of making image between style image and loss function of generating image between content image, import the neural network model for training, finally produces vivid and better visual effect image through the optimization iteration [2].

## 2 VGG -19

Realizing image style conversion requires building, a new model should be built established on the deep convolutional neural network for training. The purpose of using a pre-trained convolutional neural network is to save time and space costs. The convolutional operation is a process in which the convolutional kernel makes a small range weighted sum on each position of the input data by using the sliding window. Therefore, the convolutional operation can be popularly understood as a process of "filtering." After the interaction between the convolutional kernel and the input data, the filtered image was obtained by extracting the image characteristic. As the convolutional layer gets deeper and deeper, the higher accuracy of image feature extraction, the pre-trained VGG-19 convolutional neural network model is adopted in this paper, and its underlying architecture has been shown in Fig. 1.

To determine the style layer and content layer in the convolutional network layer and establish the loss function, it is necessary to reconstruct the feature information extracted from each segment. As shown in Fig. 2, by comparing the reconstruction results of the first layer of convolution and the original image that almost the same, also the second layer, until the results of the third and fourth layer are relatively fuzzy, the fifth layer convolutional reconstruction results can distinguish is a monkey, until the figure fc7, relu7 layer the characteristics of the original image are indistinct, cannot identify completely, the neural network learned more and more general information [3].

Comparing the reconstruct effect of each layer, it can be seen that the reconstruct effect of the shallow layer is often better. The convolutional characteristics basically retains the shape, position, color, texture and other information in the selected original image. The deep corresponding restored image loses some color and texture information, but generally retains the shape and position of the object in the original image. By comparison, we can see that for content images, the effect is inversely proportional to the depth of the convolutional layer, and the shallower the convolutional layer, the better the content characteristics in the content image can be restored. However, for style images, the deeper the convolutional layer is, the better it can restore the style features in style images.

In this paper, the content loss function of the image is established in the slightly fuzzy fourth layer, which saves a lot of high-frequency components. The image style loss function is set from segment 1 to layer 5 to ensure the image style conversion results are smoother, and the style features are prominent.

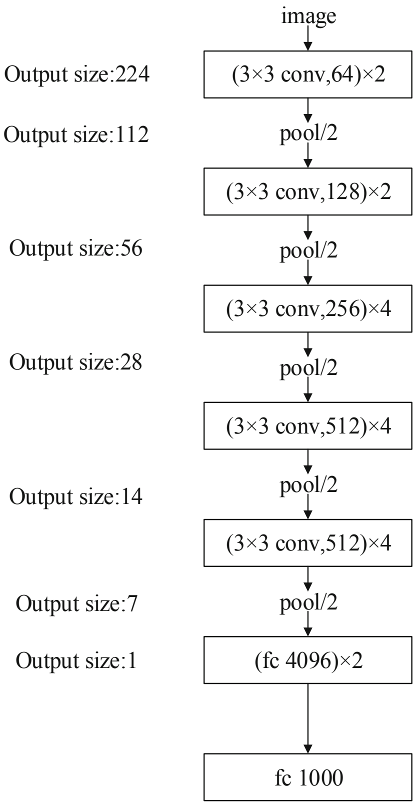


**Fig. 1.** VGG-19 convolutional neural network model

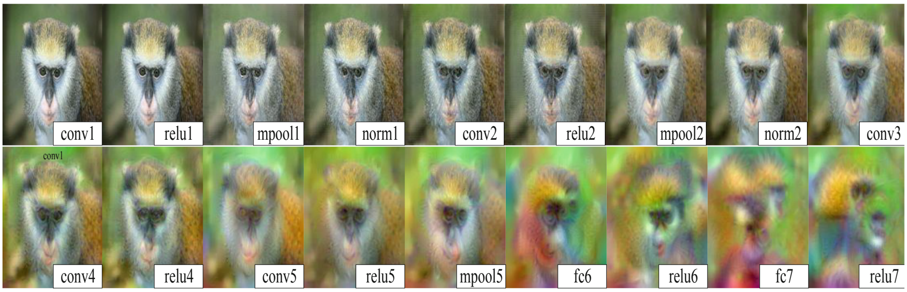


**Fig. 2.** The reconstruction of the convolutional layer feature information

# 3 Define the Loss Function

Firstly, the process of image characteristics extraction by VGG-19 convolutional network is compared with image style conversion. The purpose of the VGG-19 convolutional network is to extract the feature information from the input image and output the image category. But in the application of image style conversion, it is necessary to use the middle feature of the VGG-19 convolutional layer to restore the original image corresponding to these characteristices. In other words, the image style conversion is exactly the opposite of VGG-19. The input is these characteristics, and the output image corresponds to these characteristics.

## 3.1 Content Loss Function

The content loss Function is the difference of images content. Image characteristics are extracted from the pretraining network can be compared instead of the direct comparison between images. Features can be regarded as higher-dimensional pictures, so the better results can be achieved by using image features comparison [4].

The $C_{nn}$ represents a pre-training network that only contains the extracts characteristics by the convolution training network before. The $X$ represents an arbitrary input picture, then *Cnn (X)* can stand for a set of collections of input image's characteristics of each layer which extracted through preliminary training network, each characteristics collection has a three-dimensional matrix, let the $F_{XL} \in C_{nn}(X)$ represents the characteristics collection which has been removed from the $L$ layer of the network, its size is $h \times w \times d$, the matrix can be unfolded into a one-dimensional vector, regarded as this input picture's content of the $L$ layer in the network is $F_{XL.}$, If we need to compare the content differences between the two images, then the size of the two images should be the same. For example, $Y$ is another picture. We can define the content loss function of the two photos in the $L$ layer by using formula 1. In the equation, $F_{XL}(i)$ represents the elements of the expansion vector from the characteristics collection of the $L$ layer.

$$D_C^L(X, Y) = \|F_{XL} - F_{YL}\|^2 = \sum_i (F_{XL}(i) - F_{YL}(i))^2 \tag{1}$$

## 3.2 Style Loss Function

The definition of style loss function of the style image is not as intuitive as the definition of content loss, so we need to introduce a *Gram* matrix to represent the image style, and then calculate the style loss function. The size of the *Gram* matrix is determined by the thickness d of the characteristic graph. For each element in the *Gram* matrix, the $i$ and $j$ layers of the thick feature diagram are taken out first . In this way, two $h \times w$

matrices have been obtained, which are expressed as $F_{XL}^{i}$ and $F_{XL}^{j}$ respectively. Then, the corresponding elements of the two matrices are multiplied and summed, *Gram(i, j)* have been obtained, shown in formula 2.

$$G_{XL}(i,j) = < F_{XL}^{i}, F_{XL}^{j} > = \sum_{K} F_{XL}^{i}(k) \cdot F_{XL}^{i}(k) \tag{2}$$

$$D_{S}^{L}(X, Y) = \|G_{XL} - G_{YL}\|^{2} = \sum_{k,l} (F_{XL}(k, l) - F_{YL}(k, l))^{2} \tag{3}$$

*Gram* matrix of each element are all related to the characteristic collection of the pattern in layer *i* and layer *j*, has been expressed as the association matrix, if the *Gram (i, j)* is defined as the picture in the output of the convolution *L* network layer style, the style difference of two images can be described by the difference of *Gram* matrices, style loss function has been shown in formula 3.

## 4 Model Establishment

After selecting the content layer and the style layer, the content loss function, and the style loss function need to be added to the VGG-19 convolutional network for pre-training. The specific implementation process has been divided into two parts.

In the first part, the pre-trained VGG-19 network segment is named for comparison with the content layer and the style layer. The specific steps are as follows: Set variable $i = 0$, and then through loop iterates of VGG-19 in each layer, if condition statement judgment the layer is Conv layer, made $i = i + 1$, and the layer named conv_i, if the coating is not Conv, Elif statement whether the sheet is used to activate the sheet, if the layer called *ReLU_i*, and the sheet is not enabled, then continue to use elif statement judge the sheet is the most prominent pooling layer, f the layer called *pool_i*, if the coating is not more than three layers of cycle judgment, otherwise, Add the unique layer to the original model.

The second part, take the content loss function and loss function in content layer and style layer, steps are as follows: if statement judges the layer named in the first part with the contents of the selected layer, if in the content layer, add content after the layer loss function, and add the layer to the content of network loss, in the same way judge the named layer in the layer style, if the loss function is added after the layer style. Loss of the network will be added to the layer style. After such a complete network structure is traversed, a new network model of image style conversion is constructed, it has been shown in Fig. 3.

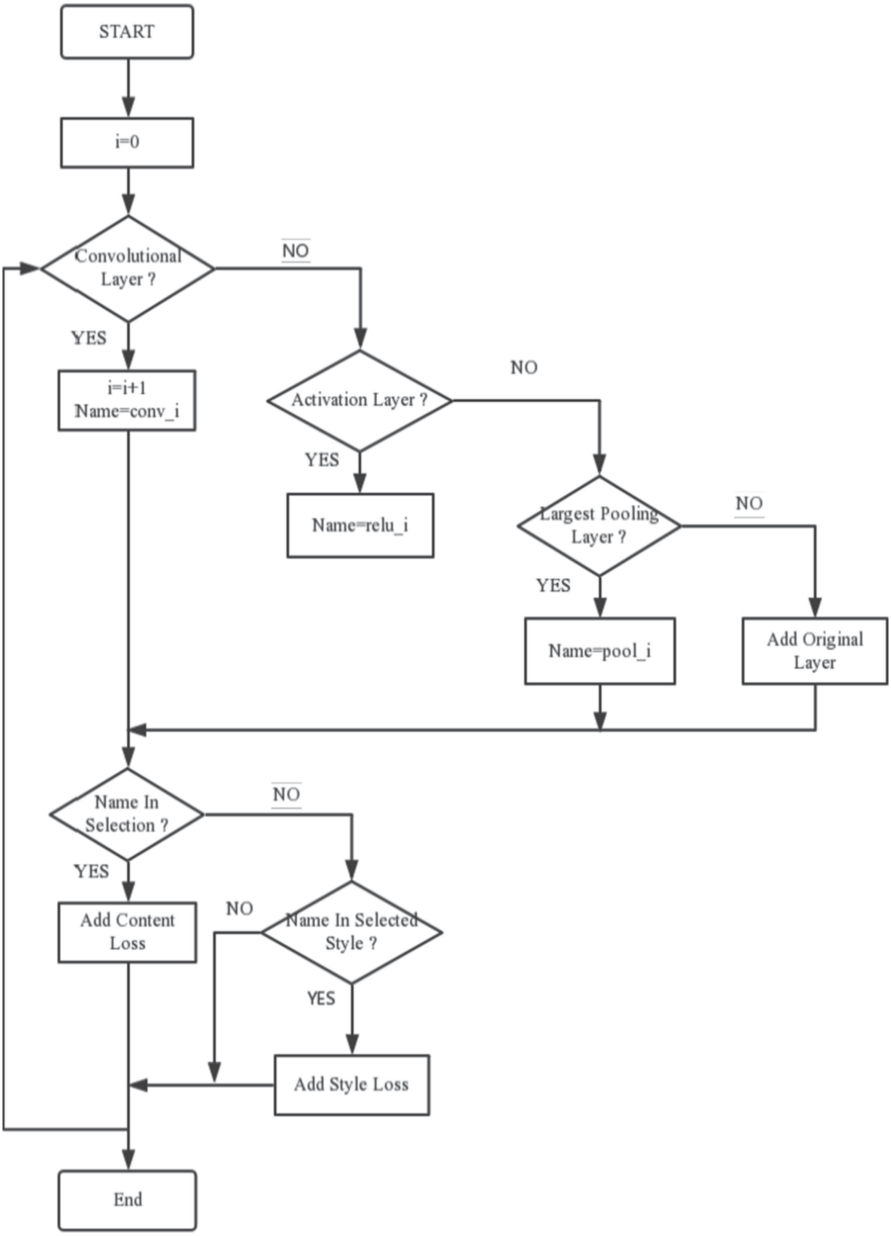


**Fig. 3.** The flow chart of image style transformation model construction

## 5 Model Training

The style conversion output image is obtained by model training. After the model training is completed, the style image and the content image are imported into the model, the style loss and the content loss are cyclically calculated, and the style loss and the content loss are weighted, and then the gradient is used. Decreasing the backpropagation update parameters to minimize the loss function, through optimization iterations, and finally output the style conversion image.

The block diagram of the back propagation gradient descent calculation loss parameter is shown in Fig. 4. First, after an image passes through the convolution layer, it is a three-dimensional array. After an image passes through the convolution layer, several three-dimensional arrays Ki are obtained. These arrays represent the content of the image. The multi-dimensional array can be used to obtain the Gram matrix operation to obtain the style of this image. After extracting the content and style of the image, the content loss and style loss are calculated**Error! Reference source not found.**. In the style transformation, content loss *DC (X,C)* and style loss *DS (X,C)* need to be minimized, so the gradient calculation of these two values needs to be carried out, the gradient calculation is shown in formula 4.

$$\nabla(X, S, C) = \sum_{Lc} w_{CL_c} \cdot \nabla^{L_c}(X, C) + \sum_{L_S} w_{SL_S} \cdot \nabla^{L_S}(X, S) \tag{4}$$

In the formula, $L_c$ and $L_s$ represent the layer output required by the content and style respectively. These parameters can be set arbitrarily according to the desired effect. The two $w$ represent the weight assigned to the content and style, respectively which can be set arbitrarily. The size of the weight will affect the degree of style conversion image.

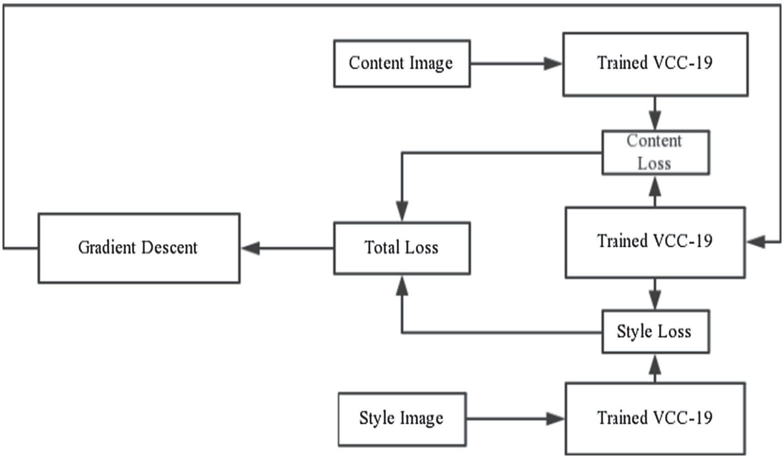


**Fig. 4.** The block diagram of reverse calculation of loss parameters

# 6 Results and Evaluation

In terms of the evaluation of the effect of image style conversion, there are two main evaluation methods, namely subjective evaluation and objective evaluation. In subjective evaluation, the main factors that lead to the effect of a picture are each person's personal preferences and evaluation methods, and can also be compared with other experimental results to analyze the quality of the experimental results. In objective evaluation, you can compare the output picture effect by comparing the size of content loss and style loss when the number of iterations is the same.

## 6.1 Subjective Evaluation of Different Weights

In this paper, the personal evaluation is obtained by adjusting the different weights in the loss function. Figure 5 is the style image, and Fig. 6 is the content image. When the weight of the loss is different, the degree of image information content and style conversion that have been retained can be controlled, so that more content or style oriented images can be obtained, shown in Fig. 7.

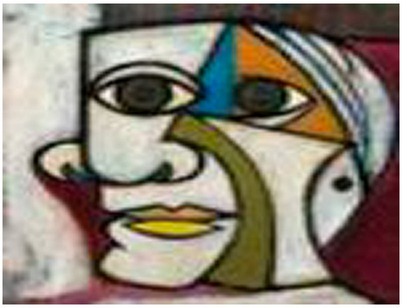


**Fig. 5.** The style image

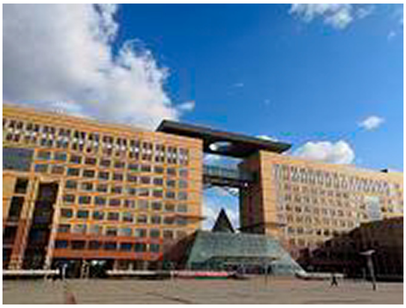
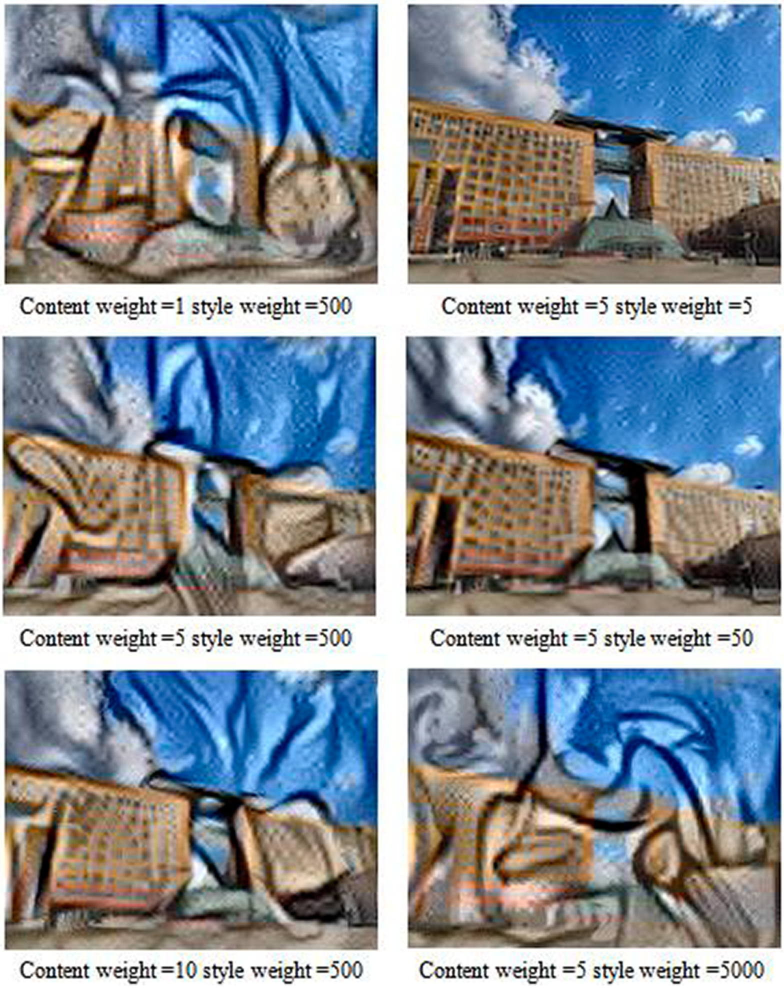


**Fig. 6.** The content image

It can be seen from the comparison that when the content weight is larger, the content characteristics of the output image are closer to the content picture, on the contrary, when the style weight is larger, the content style of the output image is closer to the style image.

## 6.2 Objective Evaluation Under Different Deep Learning Frameworks

By using two different deep learning frameworks, TensorFlow and PyTorch, under the same number of iterations, the time used and the effect of the output image were compared to achieve the objective evaluation.



Content weight =1 style weight =500

Content weight =5 style weight =5

Content weight =5 style weight =500

Content weight =5 style weight =50

Content weight =10 style weight =500

Content weight =5 style weight =5000

**Fig. 7.** The comparison of output images with different weights

Figure 10 is generated by the TensorFlow deep learning framework, and Fig. 11 is made by the PyTorch deep learning framework, both of which have 200 iterations. From the comparison of the output image effect, it can be concluded that the image color retention effect of PyTorch deep learning box is better than that of the TensorFlow deep learning frame, which retains the feeling of style image swirl more. Meanwhile, the comparison of output efficiency under the two frameworks is shown in Table 1.

It can be seen that with the same number of iterations, the PyTorch framework has a relatively small amount of computation, so it has faster operation speed and a lower CPU proportion Fig. 8 and Fig. 9.

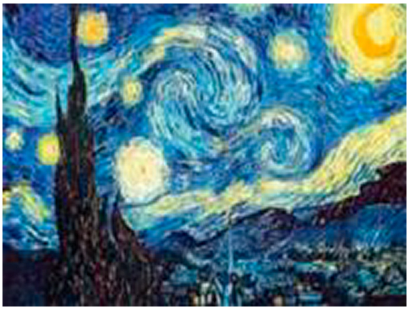


**Fig. 8.** The style image

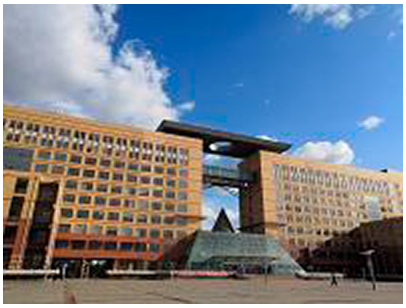


**Fig. 9.** The Content image

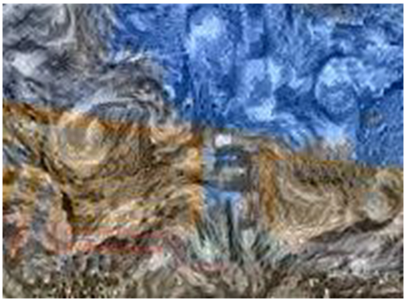


**Fig. 10.** TensorFlow output image

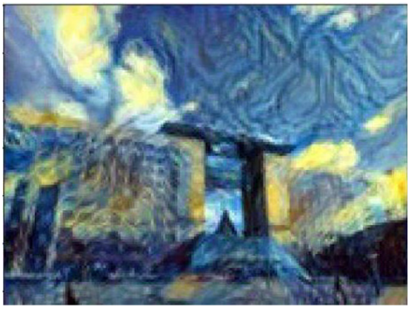


**Fig. 11.** PyTorch output image

**Table 1.** The comparison of test results

| | Time | CPU Max |
|---|---|---|
| TensorFlow framework | 313.84 s | 98% |
| PyTorch framework | 186.87 s | 70% |

## References

1. Yaling, D., Wubin, Z., Renhuang, J.: Image style transfer technology based on convolutional neural network. Mod. Comput. **630**(30), 49–53 (2008)
2. Wuyang, L.: Shallow theory of image style transformation based on deep learning. Digit. Commun. World, (2) (2018)
3. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
4. Gatys, LA., Ecker, AS., Bethge, M.: Image style transfer using convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016)
5. Guohui, F.: Classification of small-scale images based on VGG model of convolutional neural network (2018)