Partial Style Transferring and Feature Optimization for Style Transfer Network

Xiaochen Zhou
zhouxiaochen@wustl.edu

Abstract

Rendering the semantic content of an image into different styles is recently a hot topic in computer vision. Several previous works have solved a major limiting, which is the lack of image representations that explicitly represent semantic information and, thus, allow to separate image content from style. However, features or details for specific objects are sometimes become blurred, or even twisted, which makes it hard to recognize the original object from the style transferred image. In this paper, we design a pipeline to reinforce the performance of selected details in an image and make the specific part looks more natural than simply transfer the art style.

1 Introduction

Transferring the style from one image onto another can be considered a problem of texture transfer. The final goal for texture transfer is to synthesise a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image [8]. For texture synthesis, some powerful non-parametric algorithms can be used to synthesis photorrealistic natural features via resampling the pixels of a given source and reconstructing the images. Most previous texture transfer algorithms rely on these nonparametric methods for texture synthesis while using different ways to preserve the structure of the target image. For instance, Efros and Freeman present a correspondence map which includes features of the target image such as image intensity to constrain the texture synthesis procedure [3]. Hertzman et al. use image analogies to transfer the texture from an already stylised image onto a target image [4]. Ashikhmin focuses on transferring the high-frequency texture information while preserving the coarse scale of the target image [1]. Lee et al. improve this algorithm by additionally informing the texture transfer with edge orientation information [5].

As for data-driven method (deep learning), Gatys et al.[8] design the novel network to translate the feature into style vectors and synthesis the style vectors with content features and reconstruct the transferred image. However, the network cannot be sensitive to local features considering that the optimization is based on the global features. One method is to match the local feature with style image with specific compositing optimizer. Many tools have been developed for photographic compositing, e.g., to remove boundary seams, match the color or also fine texture. However, there is no equivalent for paintings. Say, if one perfers to smooth an object into a painting and ensure the content of the object, the options are limited. One can paint the object manually or with a painting engine but this requires time and the pipeline is not automatic. Also, applying existing painterly stylization algorithms as is also performs poorly, like style transfer network or real-time transfer, because they are meant for global stylization whereas we seek a local armonization of color, texture, and structure properties. The original optimization for transferring partial image to different style is not convincing neither.

In this paper, we will design a pipeline for style transfer which alow the users to select the objects that they require for detail saving, different style transformation and optimization. Also, to take the better advantage of our network, users can copy and paste any image from other source to the original image, and make the pasted part more natural with the pipeline. We will first implement style transfer network, and mask mapping function which aligns the cropped out image into suitable shape as the input of the network, then reshape the cropped out image to the original space and do the optimization. For the optimization, the local partial image selected by the users will generate one mask, then the feature extracted from the masked area will be used to compute the cosine distance with style image and pick up the closest style feature. Then compute the distance as the supervising signal.

The main contribution of this paper is as follows:

- Implement the style transfer network with VGG-19, where fewer style feature layers are utilized where fewer memory and graphic memory are used.
- Implement the image cropping and mask generation in homogenous domain with OpenCV.
- Implement the style matching algorithm for partial feature optimization.

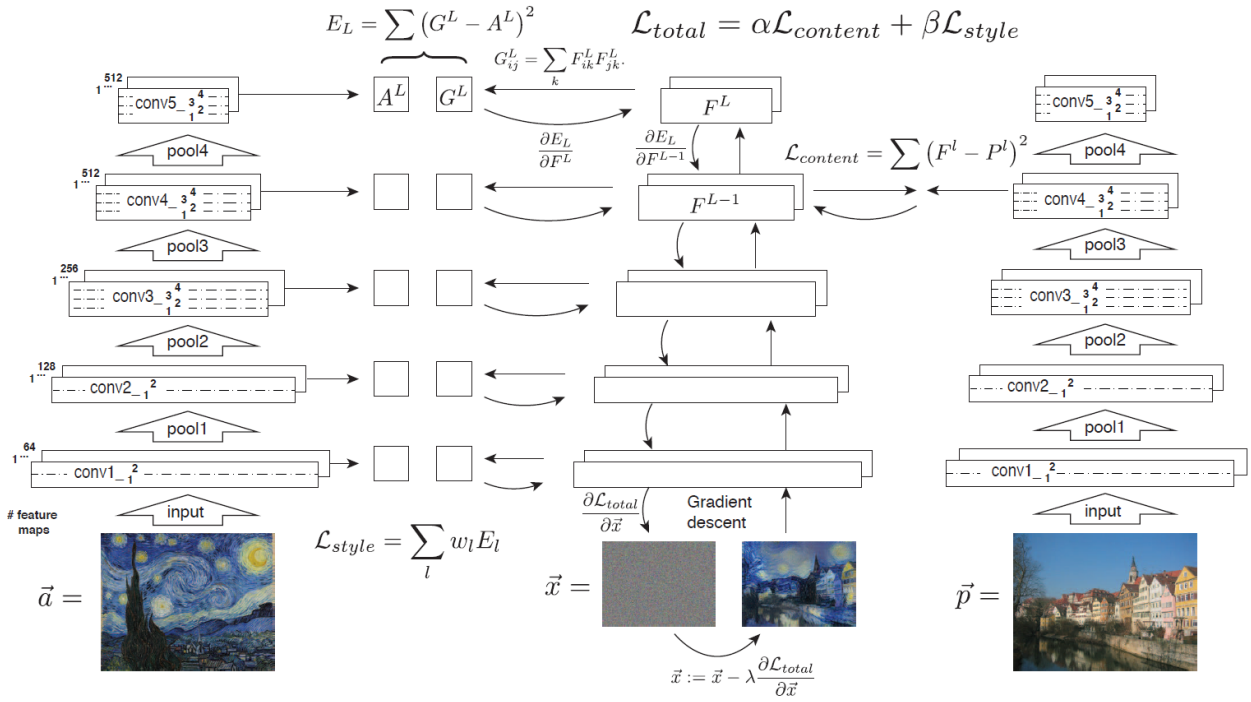

Figure 1: The structure of style transfer network

- Combine the algorithms and implement an effective pipeline for partial style transferring and pasted image harmonization.

2 Background & Related Work

2.1 Image Texture harmonization

To ease the unnatural feeling for the texture difference between patches, the simplest way is to combine the foreground and background color values with linear interpolation, which can be used for blank inpainting [2]. Using information entropy from gradient domain, or Poisson bledning, is considered an effective way, which is first introduced by Perez et al. [6]. Xue et al.[12] design an algorithm to take advantage of statistical factors like luminance, color temperature, saturation and local contrast, and match the histograms. For data-driven method, multi-scale image harmonization [7] introduced smooth histogram and noise matching while handling the fine texture on top of color features. These techniques are mostly designed for photographs in mind, however, seldom of these methods would have satisfying performance on painting harmonization. Thus, we are interested in using the advanced methods to increase the harmonization of the painting.

2.2 Convolution Neural Network

Style transferring methods would not gain such achievements without the advantage of neural network. CNNs pre-trained on large datasets like ImageNet has made great contributions to kinds of vision tasks, such as classification, retrieval, object detection etc. Utilizing pretrianed neural network lower the time-consuming of training, such as vgg-19 [11], and win good performance. In our implement, we use VGG-19 network as the baseline considering the experiment for the improving structures.

2.3 Style Trasnfer Network

Style transfer network with CNN like [8] shows impressive results on style transferring on painting task. These methods transfer arbitrary style from one image to another and keep the main structure of the original image at the same time. Recently, feed-forward generators propose fast approximations of the original neural style formulations to achieve real-time performance. Luan et al.[9] limits the mismatching of style and content image with scene analysis. Li and Wand [10] use nearest neighbour correspondences between neural responses to make the transfer content-awareness. However, these methods pay more attention on global optimization, which do damage to the local details of content information. In this case, we will find the local optimization method for the detail informa-

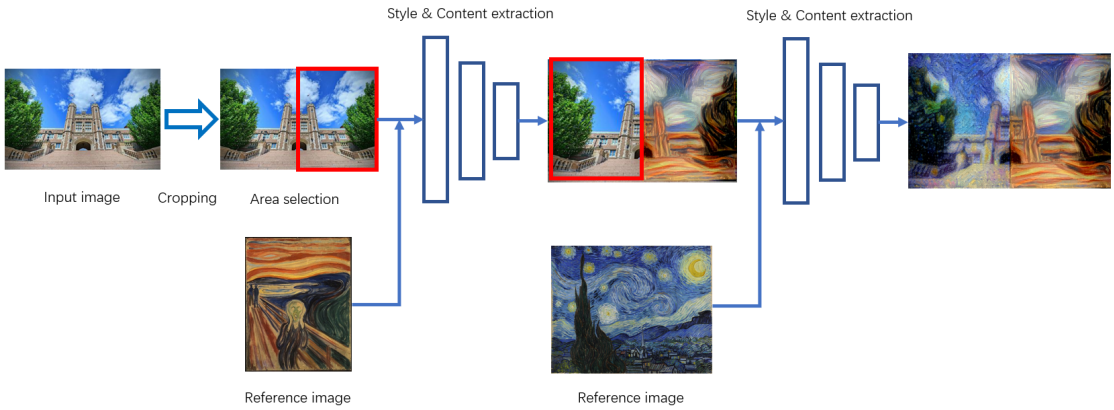

Figure 2: The structure of partial style transfer network. The red box represents the selected area.

tion and painting harmonization.

3 Method

3.1 Overview of the Pipeline

The structure of the pipeline is as follows. Figure 2.2 and Figure 3.1 show two structures of the pipeline. For both two applications, first, we need to crop out an area from the source image. This area can be the partial area which will be changed to a different style in partial transferring, or the object which will be pasted to another target image.

For partial transferring, we will align the cropped area into regular shape, making it possible for network training and generation. Then the source image and the aligned image will go through the network with different style reference respectively, generating two images. Then we realign the cropped image and paste it the source image, and smooth the edge of the cropping area.

For image harmonization, given the cropped-out local part from the source image with the object that the user require to be more natural, we will paste it to the target image and at the same time generate the mask for the pasted part. Then, we match the distance between features from masked part in target image and the style features from the reference image. Next, we will compute the distance for content vector and style features with the same method in [8], and use these evaluation value for optimization.

3.2 Style Transfer Network

Style transfer network is a structure for image reconstruction. The input of the network are the content image which provides the basic content information of the output, and the style image which supports the style feature. The network does not require the train process for the network, using the pre-trained VGG-19 network in ImageNet dataset instead.

For the reconstruction process, two images and one blank image are input into the network, where we select three layers as the style features and one layer as the content feature. Then two loss functions are used to compute the distance and reconstruct the image in the blank image, where the style feature extracted from style image and blank image are compared with Gram matrix and the distance content distance between content image and blank image are computed. The generated image will change the blank image and implement the transferring again.

After we compute the evaluation loss $L_{content}$ and L_{style}, we will add them together with weight and using this loss function as the optimizer for image reconstruction.

3.2.1 Content Representation

Given the output of the convolutional neural network, the structure feature has been encoded as a low dimension feature. In this case, we can compare the distance between the blank image with squared-error loss, which is,

$$L_{content} = \frac{1}{2} \sum ||F_{content} - F_{blank}||_2^2$$

where $F_{content}$ and F_{blank} denote the features extracted through the selected content feature layer respectively. When CNN are trained on object recognition, they develop a representation of the image that makes object information increasingly explicit along the processing hierarchy. Therefore, along the processing hierarchy of the network, the input image is transformed into representations that are increasingly sensitive to the actual content of the image, but become relatively invariant to its precise appearance. Thus, higher layers in the network capture the high-level content in terms of objects and their arrangement in the input image but do not constrain the exact pixel values of the reconstruction very much. This

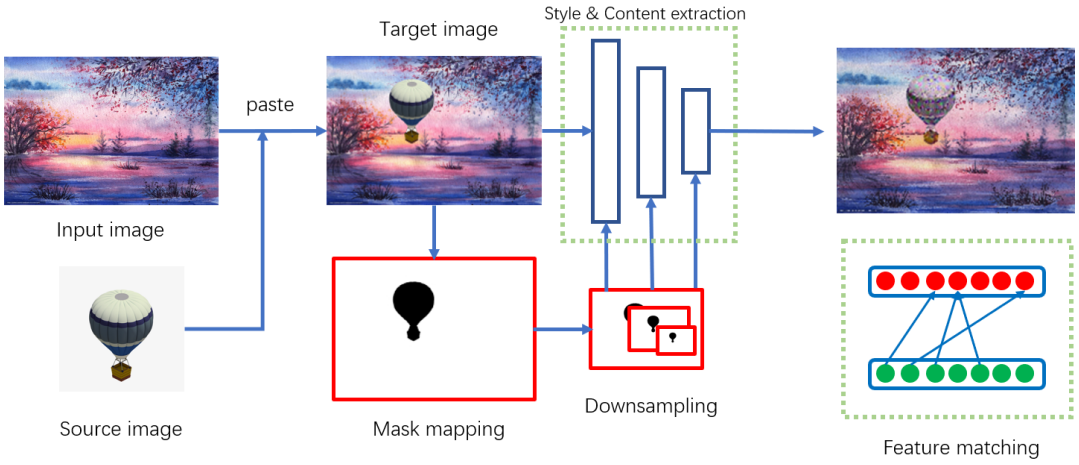

Figure 3: The structure of paste image harmonization network.

conclusion is given by Gatys et al. In this case, we will select a relatively higher layer as the content layer.

3.2.2 Style Representation

Feature space can be built on top of the filter responses in any layer of the network. It consists of the correlations between the different filter responses, where the expectation is taken over the spatial extent of the feature maps.The style representation is given by Gram matrix $G^l \in R^{N_l \times N_l}$, where $G^l_{i,j}$ is the inner product between the vectorized feature maps i and j in layer l, which is:

$$G^l_{i,j} = F^l_i F^l_j$$

where F denotes the vectorized features, k denotes the number of style layers. Then the distance between the blank image and the style image will be computed as the evaluation function:

$$L_{style} = \sum_l ||G^l_{style} - G^l_{blank}||^2$$

3.3 Partial Object Optimization

To achieve the higher performance of details in partial object, we will crop out the selected object from the original input image, paste it to the base image and optimize the output. Considering that we have achieve the style transferred background image, we only need to make the pasted part look more natural, which needs to generate the mask for the pasted part.

We use floodfill method in openCV to select the masking area and normailze it. To make the edge of the object more natural, we smooth the edge of the mask with Gaussian filter. Then we save the cropped area and paste it to the base image. In this case, we have got the materials for the optimization, the original image, the base image which is considered as the style image, the target image which is the base image with the pasted area, and the mask mapping. To optimize the target image, we input the original image and target image into the style transfer network and compute the partial difference in content feature and style feature with the downsampled mask mapping. After we get the evaluation from content loss function and style loss function, we optimize the target image only in the masked area and input the optimized image again.

3.3.1 Downsampling of Mask Mapping

To avoid the changes of the background and optimize the style in different layers, we need to downsampling the mask mapping, shaping the size of the mask the same as the feature. Mask mapping will not be considered as the input. Instead, to ensure the same dimension of the feature and mask, mask mapping will downsample one time when the input features go through one convolution block. For convolution layers, mask mapping will be resized to half of its original size. For pooling, mask mapping will be pooled with the same method in the network. Then for each feature extracted from layers, we element-wise product the mapping with features, and using these masked features for the following steps.

3.3.2 Style Matching with Base Image

Instead of computing the style difference in the global way, to make the pasted part more natural, we will compare the style feature between style features and target features pixel by pixel. To be specific, for each pixel of the two input image in the mask, we find a patch $P_l(i,j)$ sized as $d \times d \times ch$ which represents the feature of style, where d denotes the size of the patch set by users, i, j denotes the

Figure 4: Result for partial style transferring. The screen of the phone is selected as the cropping area.

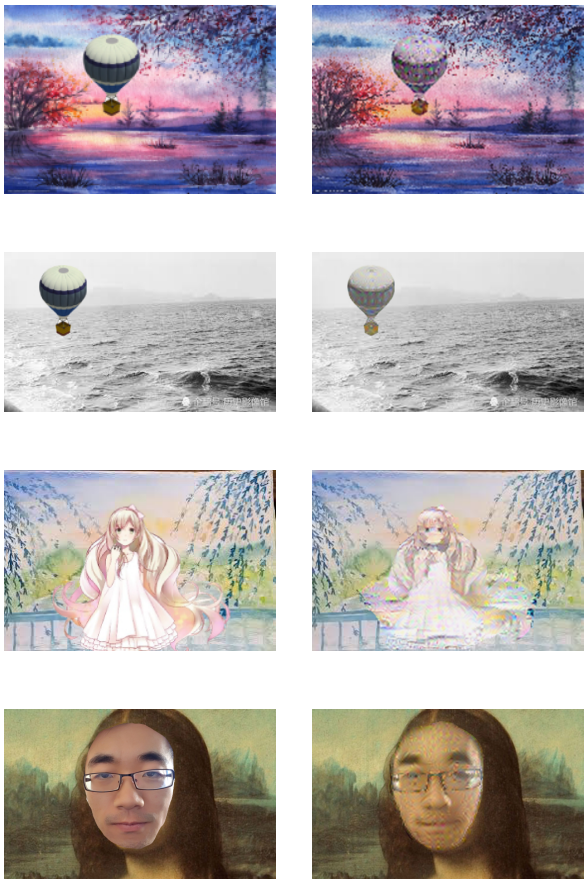

Figure 5: Experiments on pasted image harmonization. The left one is the target image (input) and the right one is the output of the pipeline.

position of the pixel and ch denotes the channel number in that layer. We find all the patches in style features and target features, align these features into vectors for Gram matrix, then match the closest vectors between patches in style features map and target features map, and compute the Gram distance between these patches, which is:

$$Loss_{style} = \sum_{l} \sum_{p} \min_{p'} ||G^{l}_{F_p} - G^{l}_{F_{p'}}||^2$$

where l denotes the layer where features are extracted, p denotes the position of feature patch in target feature, p' denotes the position in style feature, G denotes the Gram matrix, and F_p, $F_{p'}$ represent the feature vector from two feature maps patch P_l and P'_l respectively. To ease the computation and save memory, we find the matching of vectors via cosine distance first, then compute the Gram matrix distance using these matched vectors.

4 Experimental Results

4.1 Implement

This section describes the implementation details of this pipeline. We utilize VGG-19 pre-trained on ImageNet as the feature extracting network. For content feature layer, we use 'block4_conv1' considering that features extracted from this layer will not be over encoded, and the dimension of output is acceptable for our equipment. For style feature layers, 'block3_conv1','block4_conv1', 'block5_conv1' are used for feature extraction. Considering that the loss function for the optimization is organized via content loss and style loss, the weight for style loss

Figure 6: The materials for pasted image harmonization. The first one is the background image, the second one is the pasted image, or the target image, the third one is the mask mapping and the last one is the output.

is set to 1 and weight for content loss is 0.05. L-BFGS solver is used for optimization. Owing to the limitation of memory, we have to resize the input to half of the original size. Our algorithm is implemented in Keras + CUDA on GTX 1060 6G.

4.2 Application for the Pipeline

Figure 7: The comparesion between our pipeline and the original style transfer network. The left one is the target image, the mid one is from the original style transfer network and the right one is the output from our pipeline.

This pipeline has two main application. First, the user can crop out an area in the original image and transfer the style to different resource. Second, the user can copy and paste one part of image to the target image and make the pasted target image look more natural.

4.2.1 Partial Style Transferring

This application acquires the user to crop out one area and select one style image. Then our algorithm will align the cropped image to a suitable shape which can be used as the input of the style transfer network. After we get the style-transferred image, we re-align the image to the cropped shape and paste it to the original image. Figure 3.3 shows the result of partial style transferring.

4.2.2 Pasted image harmonization

This application needs the user to select an area from the input image and paste it to the target image. Our algorithm will generate the mask map and make the pasted part more natural. Figure 3.3 shows the images generated through our pipeline.

To be specific, Figure 3.3 shows the materials for pasted image harmonization. As can be seen, the pasted image, which is the shield, is unnatural in the painting. After going through the pipeline, the pasted part is much better compared with the original one.

To compared with the image generated through original style transfer network, Figure 4.2 shows the result of the same image. It is clear that the original network contains texture from the cropped area, while the image from our pipeline is much more natural, which proves that our pipeline has a stronger capability for image harmonization. (The noise comes from the limitation of memory, which will be discussed in the next section).

4.3 Discussion on the Weakness

The performance of the pipeline is satisfying, though, there are some weaknesses that can not be neglected.

One weakness is that paste image reconstructed through the pipeline will have some artificial texture and noise. These texture and noise are not expected and make the output not that natural, which is shown in Figure 4.2. We can use Gaussian filters or other methods to ease this pain, but it will blur the details in the selected area. This may come from two reason, one is downsampling algorithm for mask mapping still requires optimization, which is, methods like simple average pooling and resize the shape as half of the input are not convincing. The other is, we resize the size of the input as half of its original size owing to the limitation of memory, and enlarge the size for the output, where the low-level features in the training process are enlarged and simple interpolation for enlarge, or resize the output is not enough.

Another weakness is that the algorithm for patch match-

ing is not efficient enough, which increases the time-consuming of target image generation compared with original style transfer network. We will try to find some new method to handle these problems in the future work.

5 Conclusion

In this paper we implement a pipeline for partial style transferring and local feature harmonization. We use mask map to target the selected local part and match the patch in selected area with the style image to solve the detail blurring owing to the global optimization in style transfer network. Our pipeline has two main applications. The first is to generate multi-style image, and the second is to alow users to paste one part of the image into the target image and make the paste part in target image more natural.

References

[1] N. Ashikhmin. Fast texture transfer. IEEE Computer Graphics and Applications, 23(4):38–43, July 2003.

[2] Richard, Chang M K Y S. Fast digital image inpainting[C]//Appeared in the Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain. 2001: 106-107.

[3] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive tech niques, pages 341–346. ACM, 2001.

[4] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 327–340. ACM, 2001.

[5] H. Lee, S. Seo, S. Ryoo, and K. Yoon. Directional Texture Transfer. In Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering, NPAR '10, pages 43–48, New York, NY, USA, 2010. ACM.

[6] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. In ACM Trans. Graph. (2003), vol. 22, ACM, pp. 313–318.

[7] SUNKAVALLI K., JOHNSON M. K., MATUSIK W., PFISTER H.: Multi-scale image harmonization. In ACM Trans. Graph. (2010), vol. 29, ACM, p. 125.

[8] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[9] LUAN F., PARIS S., SHECHTMAN E., BALA K.: Deep photo style transfer. In The IEEE Conference on Computer Vision and Pattern Recognition.

[10] LI C., WAND M.: Combining markov random fields and convolutional neural networks for image synthesis. In The IEEE Conference on Computer Vision and Pattern Recognition

[11] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

[12] XUE S., AGARWALA A., DORSEY J., RUSHMEIER H.: Understanding and improving the realism of image composites. ACM Trans. Graph. 31, 4 (2012), 84.