Continuous Stroke Size Control in Style Transfer

Lingjia Meng, Yu Qin

January 2019

Abstract

The Fast Style Transfer methods have been recently proposed to transfer a photograph to an artistic style in real-time. Our work focus on controling the stroke size in the stylized images, which remains an open challenge. In this paper, we propose a stroke controllable style transfer network which can achieve continuous stroke size control. By analyzing the factors that influence the stroke size, we find out that the receptive field and the style image scales are two key factors which influence stroke size a lot. We propose a Stroke Pyramid module to produce adaptive receptive fields in our network, and two training strategies to achieve faster convergence and augment new stroke sizes upon a trained model respectively. By combining the proposed runtime control strategies, our network can achieve continuous changes in stroke sizes and produce distinct stroke sizes within the same output image.

Keywords: Neural style transfer, Continuous stroke size, Adaptive receptive fields.

1 Introduction

Re-drawing an image with a particular style has been attracting a lot attention of computer science researcher since 1990s. There are plenty works studying on texture synthesis [3, 4]. Inspired by the exciting advances in Convolutional Neural Network, Gatys explored how to use CNN to reproduce famous painting styles on natural images [5, 6]. However, the algorithm of Gatys *etal.* is based on iterative image optimization and leads to a slow optimization process for each pair of content and style. To tackle this issue, several algorithms have been proposed to speed up the style transfer process, called the Fast Style Transfer in the literature[7].

Current models of fast style transfer can be divided into 3 classes, Per-Style-Per-Model[11], Multiple-Style-Per-Model[15] and Arbitrary-Style-Per-Model[8].None of this methods can achieve continuous stroke size control.

In our project, we would like to propose a method which can incorporate multiple stroke sizes into one model and achieve continuous stroke size control. After analyzing the factors that influence the stroke size in result images, we find out that the receptive fields size and the style image scale. So we decide to establish a stroke branches module to enable our model can learn different stroke size with different receptive field. Then we use a progressive traning strategy to make the network converge faster and an incremental training strategy to learn new stroke sizes upon a trained model.

2 Background

The style of the image is difficult to describe, also is different to everyone. Some things like style and stroke are probably not defined well by artists. How to convert one style of image into another is more difficult to define,not mention to a control for style transfer, it is a nightmare for programmers.

Before neural network were introduced to this field, All the methods in this field are to create a mathematical or statistical model of a style, and then change the target image so that it can better conform to the established model [9].For example, transfer a image from day to the night. it works well, but an model only can do on one style or one picture.

After Gatys work[5, 6], we can do much more better on it, but there are still many problems in it: a very slow training process, implausible feature mixtures, style is very monotonous, there many significant work on it like Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis to reduce implausible feature mixtures[10], a fusion of different artistic style [2]. But how to control the stroke size of a style transfer process haven't been researched widely. That's just we want to do.

3 Related Work

Regulating receptive field The receptive field is one of the basic concepts in convolutional neural networks, which refers to a region of the input image that one neuron is responsive to. It can affect the performance of the networks and becomes a critical issue in many tasks (e.g., semantic segmentation [16], image parsing). To regulate the receptive field, [14] proposes the operation of dilated convolution, which supports the expansion of receptive field by setting different dilation values and is widely used in many generation tasks. Another work in [1] further proposes a deformable convolution which augments the sampling locations in regular convolution with additional offsets. Furthermore, Wei *etal*. [13] propose a learning-based receptive field regulating method which is to inflate or shrink feature maps automatically.

4 Motivation

There is a trade-off between efficiency and quality for all such Fast Style Transfer algorithms. In terms of quality, PSPM is definitely the best one, but it is not flexible in term of controlling other factors(*e.g.* stroke size, color control, etc). Different artists have different sizes even if they paint the same texture. So stroke size control is significant. If we want to derive multiple stroke sizes in one single model, resize input images might be a solution, but leads to bad results, cause it will hurt the quality of output images. So we want to achieve continuous stroke size control without trading off quality and efficiency.

5 Problem Formulation

Assume that $t_i \in T$ denotes the stroke size of an image, T denotes the set of all stroke sizes, and I^{t_i} represents an image I with stroke size t_i. Then the problem in our project can be formulated to incorporate different stroke size t_i into one single model. First, the feed-forward stylization process is as below:

$$g(I_c) = I_o, I_o \sim p(I_o|I_s, I_c) \tag{1}$$

where g is the generator. Our feed-forward style transfer process for producing multiple stroke sizes can be model as:

$$g^{'}(I_c, t_i) = I_o^{t_i}, I_o^{t_i} \sim p(I_o^{t_i}|I_c, I_s, t_i) \tag{2}$$

We aim to enable one single generator $g^{'}$ to produce stylized results with multiple stroke sizes $t_i \in T$ for the same content image I_c.

6 Proposed Methods

6.1 Analysis

Let's review the definition of stroke. Consider an image in style transfer as a composition of a series of small stroke textons, which are referred as the fundamental geometric micro structure in image. In the deep neural network based Fast Style Transfer, three factors are found to influence the stroke size, namely the scale of the style image [12], the receptive field in the loss network [7], and the receptive field in the generative network.

6.2 Network Architecture

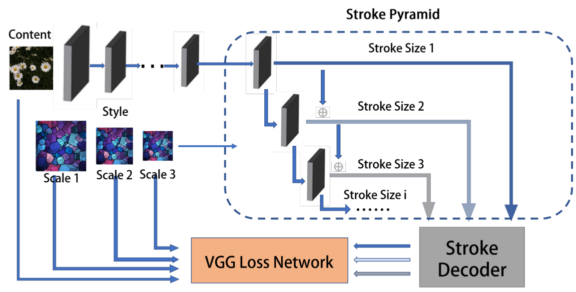

Figure 1: Model

Based on the analysis, we propose to design a network with adaptive receptive fields and each fields is used to learn one stroke size. The network architecture is depicted in 1.

There are three components in our network. The critical part is the stroke branches module. The module consists several branches and the receptive fields of each branch is bigger than the previous one through progressively growing convolutional filters. With this module, the network learns to paint picture with different stroke size.

Besides, there are another two parts of our network, namly pre-encoder and stroke decoder. The pre-encoder is the first few layers in the network and is shared by the stroke branches to learn the same semantic content of a content image and the very basic style information of style image. As for the stroke decoder, its job is to get the feature maps of the branches and then combine the stroke feature with the content semantic information to obtain the result output images with corresponding stroke size. For this module, we design a gate function G for each branch.

$$G(F^{B_i}) = a_i F^{B_i}, \sum_i a_i = 1 (a_i \in [0,1]) \qquad (3)$$

F^{B_i} denotes the feature map of branch B_i in stroke branches module, which corresponds to the stroke size t_i. When it comes to selection of a_i, in training stage, a_i is binary while at testing stage, a_i can be any rational number between 0 and 1. a_i is basis of our continuous stroke size control.
All branches will be the input of decoder. And then the decoder would decoded them into the output result with desired stroke size t_m:

$$DECODER(\sum_i G(F^{B_i})) = I_o^{t_m} \qquad (4)$$

6.3 Loss Function

Stroke Loss. The style statistics can be well represented by the correlations between filter responses of the result of the style image in different layers of VGG network. We can get these correlations by computing $Gram$ matrix over the feature map at a layer in VGG network. We use L_s to represent stroke size. Reshape $F^l(I_s)$ to $F^l(I_s)^{'} \in R^{C*H*W}$.

$$Gram(F^l(I_s)^{'}) = [F^l(I_s)^{'}][F^l(I_s)^{'}]^T \qquad (5)$$

$$L_s = \sum_{l \in l_s} ||Gram(F^l(I_s, t_k)^{'}) - Gram(F^l(I_o^{B_k})^{'})||^2 \qquad (6)$$

I^{B_k} represents the output of the k-th stroke branch. l_s is the set of VGG layers used for style loss.
Semantic Loss. We define semantic loss to preserve the semantic information of content image. We just use Euclidean distance between the content image I_c and the result image I_o in the feature space of VGG network.
Assume that $F^l(I) \in R^{C*H*W}$ represents the feature map at layer l in VGG network with image I, where C, H and W denote the channel numbers, the height and the width of the feature map respectively. So the loss is defined as below:

$$L_c = \sum_{l \in l_c} ||F^l(I_c) - F^l(I_0)||^2 \qquad (7)$$

Then the total loss function for branch B_k can be written as:

$$L_{B_k} = aL_c + bL_s + cL_r \qquad (8)$$

a, b and c are balancing factors. L_r is the regularization loss to encourage smoothness in final result.

6.4 Training Strategy

Progressive Training. To train different stroke branches in our network, we propose a progressive training strategy. The latter stroke branch benefits from knowledge of the previous branch. Suppose that the number of stroke sizes to be learn is M. For every M iterations, the network first updates the first stroke branch. Then, based on the learned knowledge of the first branch, the network uses the second stroke branch to learn the second stroke size.

Incremental Training. In order to augment a new desired stroke size upon a trained model, we use an incremental training strategy. We add one new layer as a branch in stroke branches module to get new stroke size. The location of new layer depends on the previous branch.

6.5 Continuous Stroke Size Control

In our model, we use feature interpolation strategy to interpolate between trained stroke sizes in the feature space. Consider an image I_c, we suppose that F^{B_m} and F^{B_n} are two output feature maps of stroke branches module. So a new feature map F^{B_k} can be obtained by controlling the gate function G:

$$F^{B_k} = aF^{B_m} + (1-a)F^{B_n} \qquad (9)$$

Change the value of a, continuous stroke size control can be achieved.

7 Experiments

7.1 Implementation Details

Our network is trained on MS-COCO dataset 2014. All images are resized to 512*512 pixels. We use Adam optimizer in training. We choose VGG 19 as our loss network and $relu1\_1, relu2\_1, relu3\_1, relu4\_1, relu5\_1$ are used as the style layers and $relu4\_2$ is used as the content layer. The default number of learned stroke size is 3. The scales are 256, 512, 768.

7.2 Evaluation

As you can see in Fig 2, the respective stroke size of the eight images is different. Our model is effective at producing fine stroke and preserving the semantic details.

8 Conclusion

In this paper, we explore a continuous stroke size control approach for fast style transfer. And we also talks about the adaptive receptive fields in

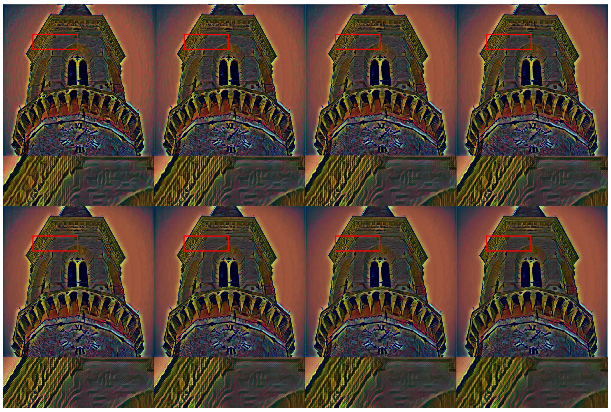

Figure 2: Results of continuous stroke control. We zoom in on the same region for better observation of the variations of stroke sizes.

CNN, which we consider as an important part of this project. In the future, we hope to apply it to larger research areas.

Our work is only a small step towards flexible stroke size control. There are few problems remaining to be explored. One that we are most interested in is the relations among the style representation of different scales of the same style image. We will do more research later.

References

[1] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017.

[2] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *ICLR*, 2017.

[3] Michael Elad and Peyman Milanfar. Style transfer via texture synthesis. *IEEE Trans. Image Processing*, 26(5):2338–2351, 2017.

[4] Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 553–561, 2016.

[5] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 262–270. Curran Associates, Inc., 2015.

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[7] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017.

[9] Eero P. Simoncelli Javier Portilla. A parametric texture model based on joint statistics of complex wavelet coefficients. In *International Journal of Computer Vision*, page 49–70, 2000.

[10] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016.

[12] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer, 2016.

[13] Zhen Wei, Yao Sun, Jinqiao Wang, Hanjiang Lai, and Si Liu. Learning adaptive receptive fields for deep image parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2434–2442, 2017.

[14] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions, 2015.

[15] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017.

[16] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation, 2018.