Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization

Jiadong Zhu, Yicong Tao

Based on the paper:

Xun Huang, Serge Belongie; Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization (ICCV 2017)

INTRODUCTION

Style transfer, also called texture transfer, is a common problem in image processing which can change the style of an image while preserving its semantic representation. Most previous texture transfer methods utilizes non-parametric algorithms by resampling the pixels of a given source texture. The main difference is their choices in how to preserve the structure of the target image. Although remarkable results are achieved through these methods, a fundamental limitation cannot be circumvented: they only use low-level image features which are mainly concluded from prior knowledge to guide the texture transfer. However, an ideal style transfer algorithm should be able to render the semantic content of the source image, such as objects or general scenery, in the style of the target image by extracting the structural information from the source image. Therefore, a basic prerequisite is to model image representations of the content and the style of images independently.

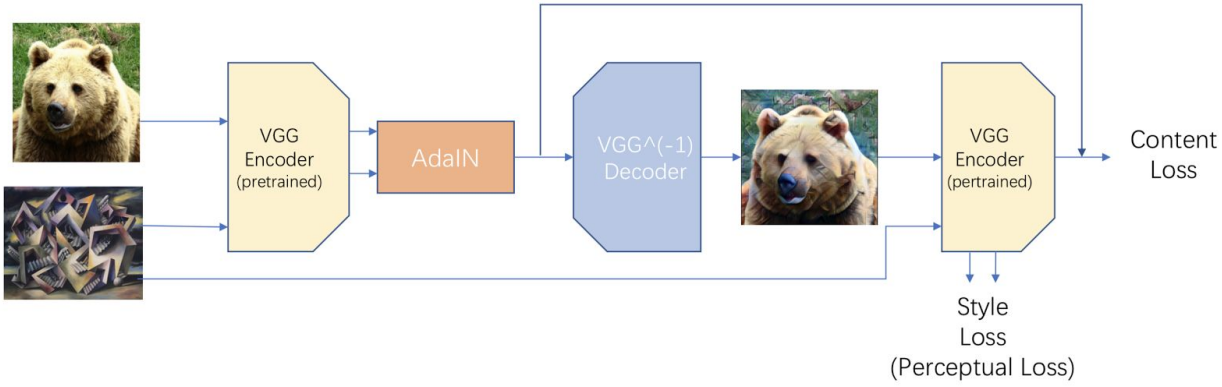
Generally speaking, such content-style separating in natural images or common photographs is still a non-trivial problem. However, the recent advancement of Deep Convolutional Neural Networks **[1]** has achieved a new way of representing images and the algorithm can learn to extract high-level semantic and style information from natural images. It was even proved that Convolutional Neural Networks can generalise across various datasets and even to other visual information processing tasks, including texture recognition **[2]** and artistic style classification **[3]**.

RELATED WORK

Gatys et al. introduced a neural style transfer algorithm that renders a content image in the style of another image using Deep Convolutional Neural Networks **[4]**. However, their framework requires a slow iterative optimization process which greatly limits its practical application. Recently, several follow-up methods such as fast approximations with feed-forward neural networks have been proposed to speed up neural style transfer. Johnson et al. **[5]** proposed a new loss function called Perceptual Loss which utilized the feature extraction capability in the Deep Convolutional Neural Networks to evaluate the content and style losses. Their model achieves multiple content images

transfer with one trained model, which accelerates the inference process dramatically. In this project, we implemented a newly-proposed method by Huang et al. [6] which presents a simple yet effective approach that can achieve real-time arbitrary image style transfer. It uses a novel adaptive instance normalization (AdaIN) layer that aligns the mean and variance of the content features with those of the style features.

MODEL



In our project, we use the model proposed by Huang et al[5]. First, the content image and corresponding style image are input into a VGG encoder. We extract features from VGG-19's relu-4-1 layer as indicated in the paper. Second, the extracted content feature Fc is aligned with style image feature Fs using AdaIN. Then, aligned content feature T is input into a deep convnet decoder and its output is the style-transformed image G.

Encoder

Along with the paper, we use ImageNet pretrained VGG-19 [7] from conv1_1 to relu_4_1 as the encoder.

Decoder

The decoder is basically a inverted VGG-19 from relu-4-1 downto the beginning, but all pool later is replaced by upsample layer. In the original paper, they use nearest upsample, but in our experiment, it can easily lead to overfitting (with artificial stroke). We found bilinear strategy comes with a better result, though not much.

Loss

The loss function consists of two parts: style loss and content loss.

Style loss is computed using perceptual loss by Johnson[5]. It's the sum of L2 loss of the VGG encoder hidden layer outputs' variance and mean from the output image G and style image S.

Content loss is directly calculated from the L2 loss of the aligned content feature and transformed image's VGG feature.

TRAINING

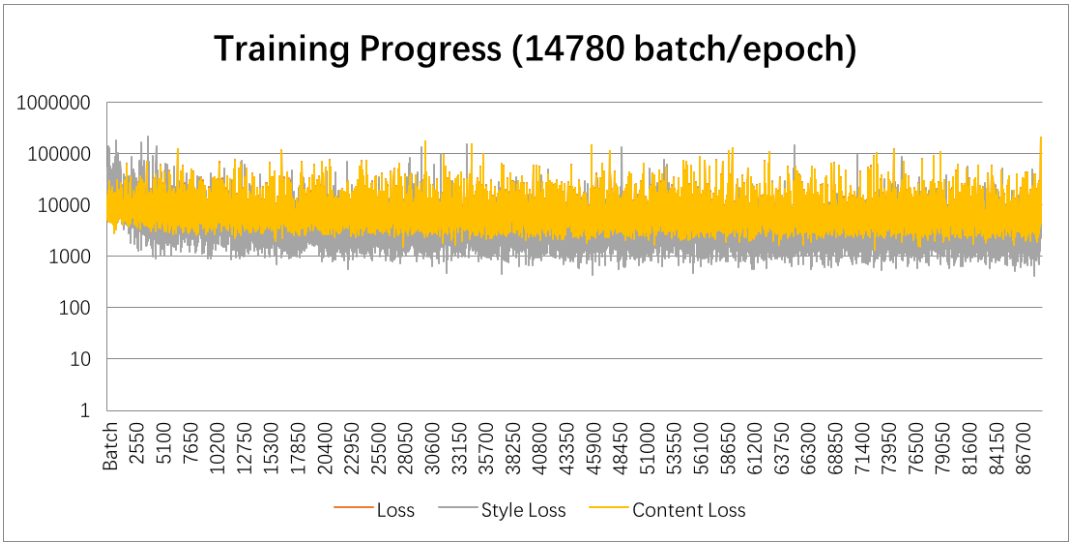
Dataset

In our training process, we use MS COCO 2017 as content image dataset, which contains 123K images for training, 41K for testing and 5K for validation. For style image, we use WikiArt Dataset (download from [8]) containing 80K style images and we divided it into 70%, 20% and 10% as training, testing and validation set.

Training Strategy

We choose to use SGD optimization algorithm, with 0.9 momentum. Every mini-batch contains 8 content images and 1 style image.

The training progress is shown below.

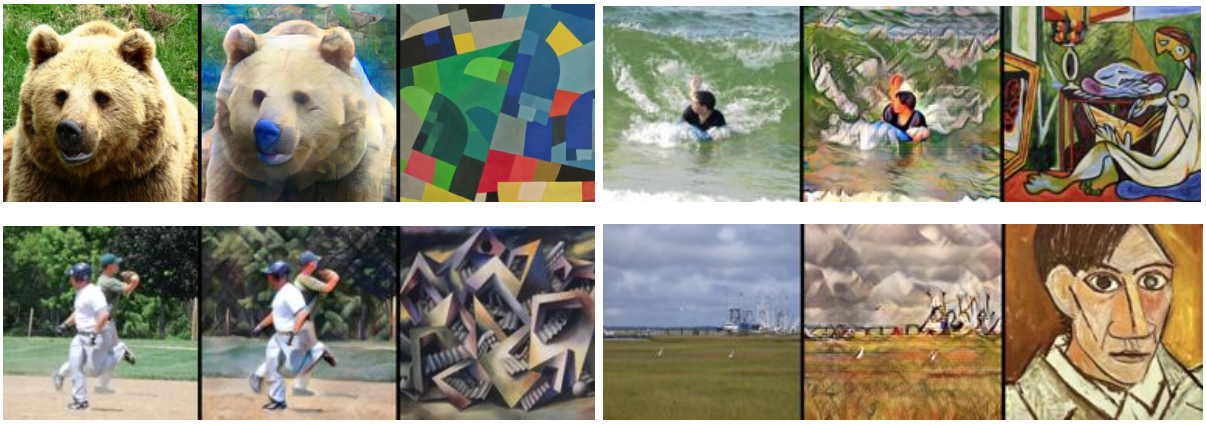


Training Progress Graph

Training output from epoch 0 (batch 720), epoch 0 (batch 12410), epoch 1 and 4

RESULTS

Some of our results on testing set is shown below. Note that, both content images and style images here are not involved in training.



We also tested different content-style trade-off in our model. Due to random crop, the output image may shift a little bit. It is apparent that as alpha increases, the stylization becomes more and more significant.

From left to right: alpha = 0.0 (no style), 0.25, 0.5, 0.75, 1.0, and the style image.

DISCUSSION

We found that the overfitting issue is kind of serious in this model. With the epoch increasing, the artificial stroke becomes more and more significant. We believe it is probably due to too many impressionism style images in the training set, making the decoder leaning towards generating more brush strokes. In the next step, we will try to reduce them in the training set.

On the other side, we found there is another paper [9] also very interesting, in their model, the decoder is only trained with content images. They use "whitening and coloring transforms" to directly transfer features. We believe this model share some ideas with Huang's model, but since it does not need to be trained with style image, it has better generalization ability. We would try to implement this model in the future.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[3] S. Karayev *et al.*, "Recognizing Image Style," in *Proceedings of the British Machine Vision Conference 2014*, 2014.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," *arXiv [cs.CV]*, 27-Mar-2016.

[6] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization," *arXiv [cs.CV]*, 20-Mar-2017.

[7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv [cs.CV]*, 04-Sep-2014.

[8] "WikiArt dataset." [Online]. Available: http://www.cs-chan.com/source/ICIP2017/wikiart.zip. [Accessed: 21-Dec-2017].

[9] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal Style Transfer via Feature Transforms," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 385–395.