Image Style Transfer for Distillation of Diffusion Knowledge into a Transformer

Middle talk

Silvestrov Egor

Department of Mathematical Forecasting Methods, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University

November 22, 2024

Goals

Transferring the image style in a more general setting requires using the input image of the content X_c and the input image of the style X_s to obtain a stylized image of the content with this style $\hat{X}_t$. This task can be described as follows:

$$f_\theta(X_c, X_s) = \hat{X}_t,$$

where $f_\theta(\cdot, \cdot)$ is the model and θ is its parameters. Neural networks are used as models.
Our goals are to get an effective and lightweight solution to the problem of transferring style from an image.

The Learning Process

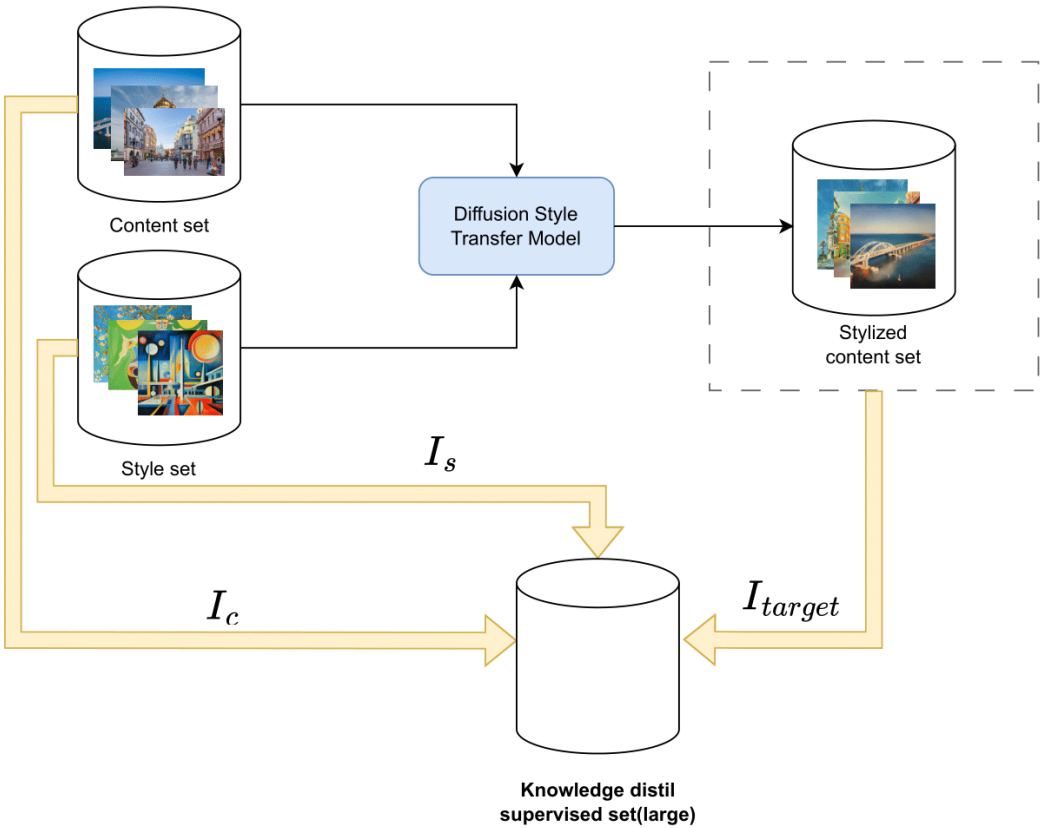

Figure: The Learning Process.

Literature

[1] Z. Wang, L. Zhao and W. Xing, "StyleDiffusion: Controllable Disentangled Style Transfer via Diffusion Models", College of Computer Science and Technology, Zhejiang University, 2023.

[2] Y. Zhang, N. Huang, F. Tang, H. Huang, C. Ma, W. Dong, C. Xu, "Inversion-based Style Transfer with Diffusion Models", MAIS, Institute of Automation, Chinese Academy of Sciences, Institute of Computing Technology, Chinese Academy of Sciences, School of AI, UCAS, Kuaishou Technology, 2023.

[3] S. Yang, H. Hwang, J. Chul Ye, "Zero-Shot Contrastive Loss for Text-Guided Diffusion Image Style Transfer", Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science and Technology (KAIST), 2023.

[4] J. Chung, S. Hyun, J. Heo, "Style Injection in Diffusion: A Training-free Approach for Adapting Large-scale Diffusion Models for Style Transfer", Sungkyunkwan University, 2024.

[5] J. Wang, H. Yang, J. Fu, T. Yamasaki and B. Guo, "Fine-Grained Image Style Transfer with Visual Transformers", The Univerisity of Tokyo, Microsoft Research, 2022.

[6] K. Simonya, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Visual Geometry Group, Department of Engineering Science, University of Oxford, 2015.

Problem Statement

Style transfer task is usually an unsupervised task and is solved with the help of specially selected loss functions.
Recently, diffusion models do a good job with this task, but they are very heavy and expensive to calculate, so in the current article this task becomes a supervised task, since the knowledge of the diffusion model is distilled into a smaller model (transformer) optimizing the loss function:

$$\underset{X_t \sim P(X_t)}{\operatorname{argmin}} \mathbb{E} L(\hat{X}_t, X_t),$$

where X_t is the result of the diffusion model, $\hat{X}_t$ is the result of the student (transformer) model and $P(X_t)$ is distribution of diffusion model results X_t.

Problem Solution

The VGG19 model was used to calculate some loss functions. 3 objective functions were tried and as a result, the MSE function gave the best results:

$$L(\hat{X}_t, X_t) = ||\hat{X}_t - X_t||_{l_2}^2.$$

MSE and the author's subjective assessment were used as criteria for evaluating the result.

Experiments

The SIID ([1]) model was used as a techaer model and STTR ([5]) model was used as a student model.
Of the popular image style transfer diffusion models [1, 2, 4], only [1,4] have been tested, since they have an implementation code in the public domain. The problem with model [2] was that it generated images with style elements that were not in the style picture, that is, the model generated new content based on its knowledge. Method [4] works more like all classical approaches of transferring the image style, trying only to extract dependencies for content from the current style image and it is better suited for distillation, therefore it is used as a teacher model.

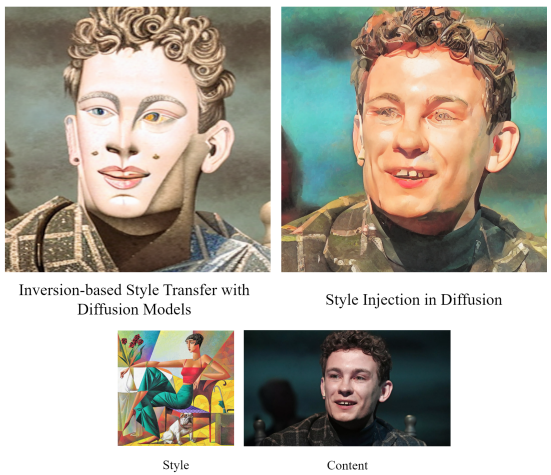

Figure: The result of comparing two diffusion models:Inversion-based Style Transfer with Diffusion Models and Style Injection in Diffusion.

Teacher's Architecture

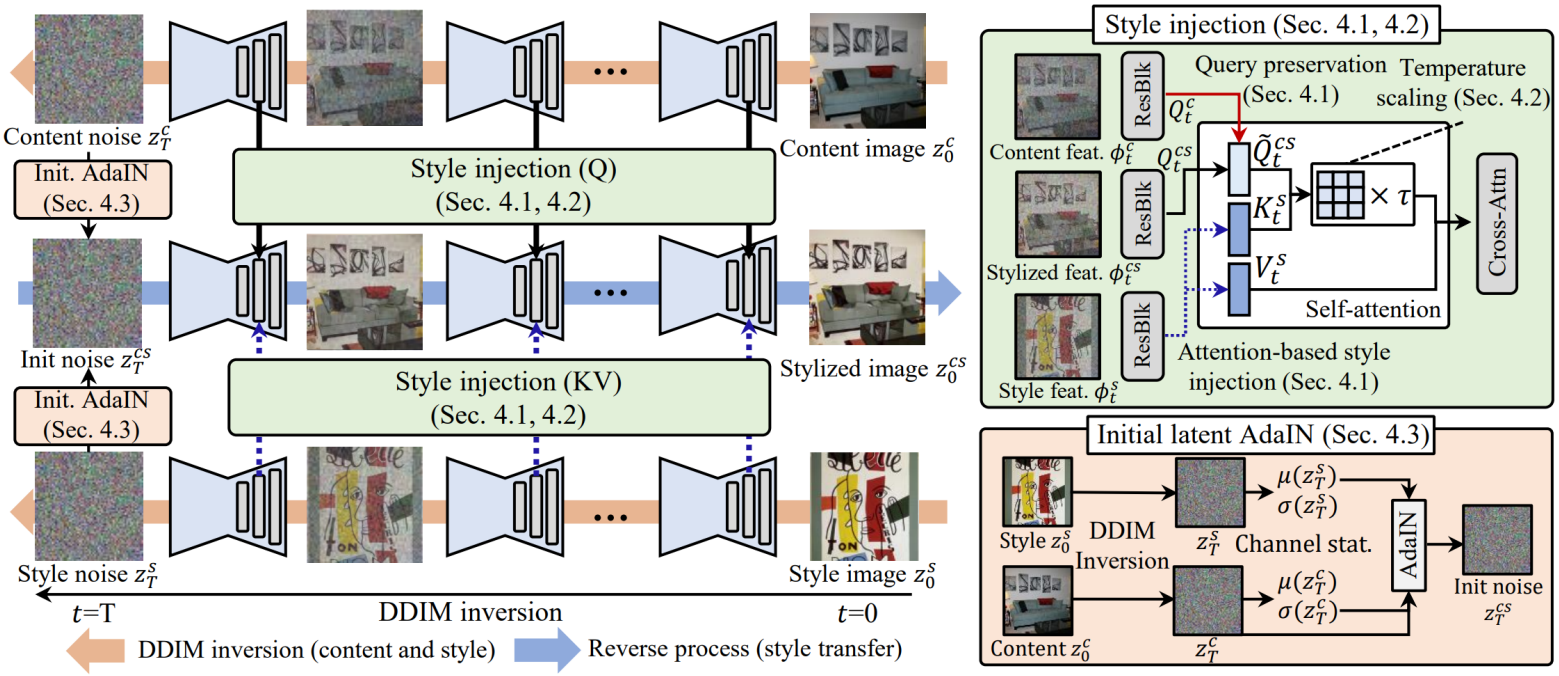

Figure: Teacher's Architecture.

Student's Architecture

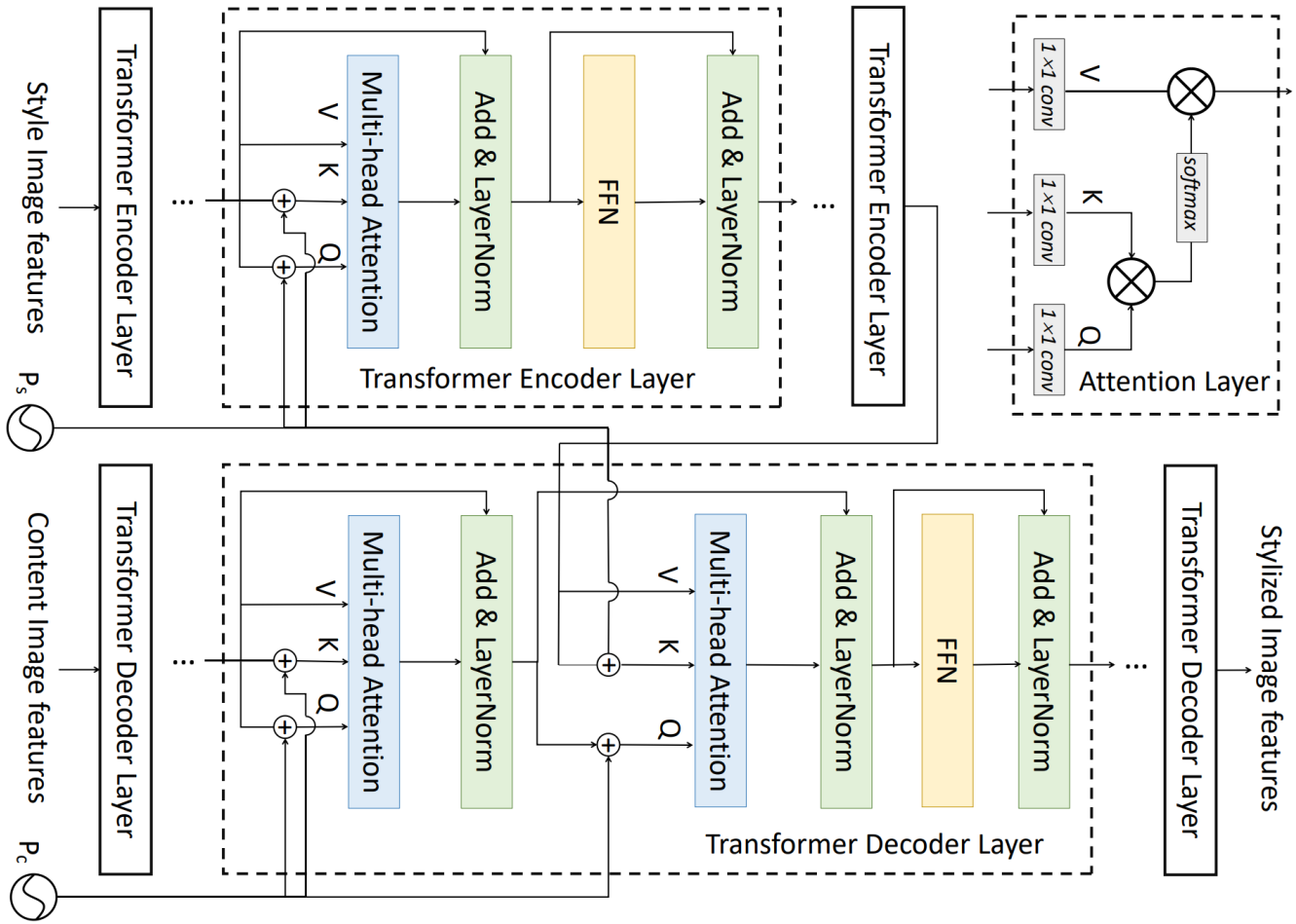

Figure: Student's Architecture.

Experiments

Several different types of loss functions were tried for experiments, but only the selected three gave significantly different results:

$$L_1(\hat{X}_t, X_t) = ||\hat{X}_t - X_t||_{l_2}^2, \tag{1}$$

$$L_2(\hat{X}_t, X_t) = 0.5 \cdot ||\hat{X}_t - X_t||_{l_2}^2 + 0.5 \cdot \Sigma_{i=1}^4 ||\Phi(F_i(\hat{X}_t)) - \Phi(F_i(X_t))||_{l_2}^2, \tag{2}$$

$$L_3(\hat{X}_t, X_t) = 0.7 \cdot ||\hat{X}_t - X_t||_{l_2}^2 + 0.3 \cdot \Sigma_{i=1}^4 ||\Phi(F_i(\hat{X}_t)) - \Phi(F_i(X_t))||_{l_2}^2, \tag{3}$$

where $F_1(X)$ is the activation slice from the 1st to the 2nd layer of VGG19, $F_2(X)$ is the activation slice from the 3rd to the 7th layer of VGG19, $F_3(X)$ is the activation slice from the 8th to the 12th layer of VGG19, $F_4(X)$ is the activation slice from the 13th to the 21st layer of VGG19, $\Phi(X)$ calculates the gram matrix for the input X.

Experiments

Some of the results of the work on the gold set can be seen in Figure 5.

Figure: Comparison of 3 objective functions on a golden dataset. SIID is Style Transfer with Diffusion Models from [4] and STTR is STyle TRansformer from [5].

Results and Conclusion

The general properties for the learning result are that the model has lost the defects of the picture and the colours of the image began to adjust to the result of the teacher's work. The content tends to be preserved, and the style is applied mainly in colour. Loss L_2 has a problem, since its result is less saturated with the colour of the style, and also has more defects than loss L_1 and L_3. L_1 and L_3 are mostly different in brightness and L_1 is brighter, so this option is the best among the others.
When distilling model [4] into model [5], defects disappear from model [4] and it transmits colours more correctly, however, the disadvantage is that it practically does not change the content for the style structure, but we got a model that weighs 10 times less and can run on a much larger number of video cards. in further studies, we want to try other architectures for the student's prayer and various variants of the object function.