Extensions to Neural Style Transfer

Team 4D Tensor
Neha Chawla, Swarnita Venkatraman, Sneha Bandi, Praveen Iyer

1 Introduction

Given a content and a style reference image, the process of applying styles to generate an output image that retains the core elements of the content image but appears to be in the style of the reference image is known as Neural Style Transfer (NST) (Gatys et al., 2015). In recent times, NST has achieved good performance, but studies highlighting improving its stability, quality, flexibility and evaluation are still in its early stages, specifically using user feedback. Besides, users may not have exact image enhancements or corrections in mind but may still be interested in enhancing their images. To build upon these challenges, we propose extension to the original NST technique to enhance the final style of the output image, using the novel pipeline of Cascade Style Transfer (CST) and User feedback mechanism. Our key idea focuses on, using CST to combine multiple style images, alongside human observed heuristic knowledge to improve stylistic quality of generated image. Our cascade framework contains two architectures, i.e., Serial Style Transfer (SST) and Parallel Style Transfer (PST) (Wang et al., 2020). For evaluation and improving the stylization quality through user preference we used three quantifiable factors - content fidelity (CF), global effects (GE) and local patterns (LP) (Wang et al., 2021).

2 Problem Definition

Multiple style transfer in NST enables styles of multiple images to be transferred to a single content image by modification of style losses. Style from the style image is known to be preserved more in CST while in traditional Multi Objective (MO) Networks content from content image is known to be preserved more (Wang et al., 2020). Our project performs multiple style transfer using CST to exploit its benefit of preserving style more, giving us room to incorporate multiple styles enabling generation of images with better quality.

Evaluating and improving the stylization quality remain two important open challenges, there are no standard quantifiable metrics defined to measure its performance. So far, most papers have used qualitative evaluation in the form of user studies for NST (Gatys et al., 2015). This can lead to subjective opinions mattering more rather than having a clear objective which is being optimized. Moreover, there is not enough literature on work done in incorporating user feedback with NST to modify/personalize the generated image based on user's preferences and doing multiple style transfer using CST.

3 Description of (tentative) solution

We propose explorations of extensions to the original NST algorithm to improve the quality of multiple style transfer. Our solution pipeline focuses on combining multiple architectures in serial (SST) and parallel (PST) as a part of cascading style transfer mechanism. We also intend on incorporating user feedback based on the results through evaluation metrics and by weighting the given styles based on their individual losses.

Based on our experiments so far, when the same architecture is used twice for CST (serially/parallely) and hence stylistic characteristics of style 2 replaces and dominates over style 1 in the stylized image. To address this we have implemented multiple architectures independently. As a tentative solution, we are planning to integrate and experiment with a combination of different architectures in our CST setting to alleviate the existing problem. For example model 1 can focus on local patterns of style 1 and model 2 can focus on global effects for style 2. We also plan on using three different evaluation metrics where each of them focuses on different aspects of the image to further enhance the output image.

4 Experiment settings

4.1 Dataset

Since NST doesn't fall under supervised learning or unsupervised learning, there is no fixed dataset that we are using. For testing and development purposes we have been using the same set of images as provided in the GitHub repository (Gordić, 2020).

4.2 Baseline Model

As a part of the baseline model (Gatys et al., 2015), style transfer is performed by using gradient descent on random noise to minimize the deviation from content of the target image. The deviation from the style representation of the style image, is defined as a set of Gram correlation matrices G_l with the entries $G^l_{ij} = (F^l_i, F^l_j)$,where l is the network layer, i and j are two filters of the layer l and F^l_k is the array (indexed by spatial coordinates) of neuron responses of filter k in layer l. We have therefore used the VGG-19 network for all our experiments. It performs better at style transfer, as its architecture captures more information compared to other networks like AlexNet, GoogLeNet which strongly compress the input at the first convolutional layer, thus losing a lot of fine details. The optimization algorithm used is L-bfgs where the max iterations was set to 1000 iterations. L-bfgs is preferred over Adam for NST because it is a second-order optimization algorithm and can handle larger batch sizes, and has been shown to work well in practice for NST.

4.3 Implementations

Initially, we used a simple NST method based on the traditional gram matrix based architecture. As a starting point, we decided to modify the network as a MO network which will be able to handle our multi style image setting. This was giving satisfactory results visually but based on our hypothesis, we wanted to experiment with CST to see if our results improve as hypothesized.

4.3.1 CST Architectures

4.3.1.1 Serial Style Transfer (SST) SST is a technique that involves applying a sequence of style transfer algorithms or models to an image (Wang et al., 2020), with each subsequent model building on the output of the previous one. We tried implementing SST which happens to be a variant of CST. So we used the same architecture serially, the stylistic characteristics of style 1 were replaced by the stylistic characteristics of style 2 on the content image. Results obtained from the SST method through cascading of the gram matrix architecture twice were not promising.

4.3.1.2 Parallel Style Transfer (PST) In PST (Wang et al., 2020), the total loss is a weighted linear combination of different losses coming from different methods connected in parallel. All the methods are optimized in parallel with the total loss. The paper mentioned PST as a method as it is known to improve the quality and flexibility of style transfer. Similar to our experiment conducted for SST where the multi-objective network was used twice, an attempt to recreate that with PST was made. However, since MO networks already share the same backbone and optimize a total loss function which is the addition of content loss, style 1 loss and style 2 loss it becomes mathematically the same as a weighted version of PST. This approach did not seem to be an efficient choice and gave rise to the need of implementing a different method to combine alongside the implemented MO method for creating the PST architecture.

4.4 Alternate Architectures

Many methods were considered for exploration based on the paper (Wang et al., 2020). However, since the paper focuses solely on single style transfer, methods specifically catered to Multiple Style Transfer were also considered.

4.4.1 Variational AutoEncoders (VAE)

ST-VAE, is a Variational AutoEncoder for latent space-based style transfer which performs multiple style transfer by projecting nonlinear styles to a linear latent space, enabling to merge styles via linear interpolation before transferring the new style to the content image (Liu et al., 2021). According to the paper this method is faster and flexible for multiple style transfer compared to baseline methods. The ST-VAE method was implemented using existing code for both single and multiple style transfer. For multiple style transfer the existing code base focuses on using four style images which was modified according to the project goals which aims at combining two style images. The method shows promising results and would be further used for combining with the gram matrix method in the PST architecture during the post-midterm phase for the project.

4.4.2 Vision Transformer (ViT)

Existing approach adopts a global feature transformation to transfer style patterns into content images. Such a design usually destroys the spatial information of the input images and fails to transfer fine-grained style patterns into style transfer results. To solve this problem, we use the STyle TRansformer (STTR) network proposed in (Wang et al., 2022) which breaks both content and style images into visual tokens to achieve a fine-grained style transformation based on attention mechanism (Deng et al., 2021).

4.5 Evaluation procedure

4.5.1 Metric 1: Content fidelity (CF)

It is a way to measure the extent to which the content of the original image is preserved in the stylized image. Since NST involves transforming the content of the original image to match the style of a style image, it is important to evaluate the quality of the output with respect to how much it retains the original content.

4.5.2 Metric 2: Global Effects (GE)

It is a measure of overall visual quality in NST that evaluates the similarity between the transferred style and the style image and it reflects the initial impression that the image leaves on human visual perception. It encompasses two aspects: global colors (GC) and holistic textures (HT), which were identified as important factors in user studies. GC compares the similarity of color histograms, while HT uses the Gram matrix to capture the holistic textures across multiple layers. The GE factor is the average of both GC and HT.

4.5.3 Metric 3: Local patterns (LP)

It is an evaluation metric to measure the extent to which patterns like brush strokes and exquisite motifs from the style image are transferred to the stylized image. This is done by using two components: the first assesses the similarity between the local patterns in the original and stylized images, while the second compares the variety of the pattern categories identified in both images.

5 Results and Discussion

As shown below, based on the inputs used in 1 the left image in 2 shows visually satisfactory results for the MO network. The right image for SST shows how style2 dominates over style1 which is a result of using the same architectures for SST.

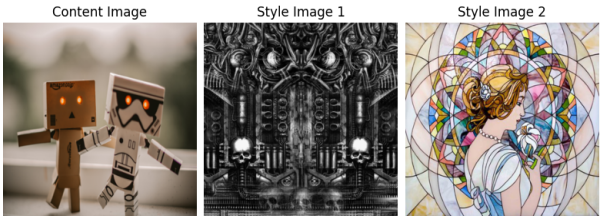

Figure 1: Inputs

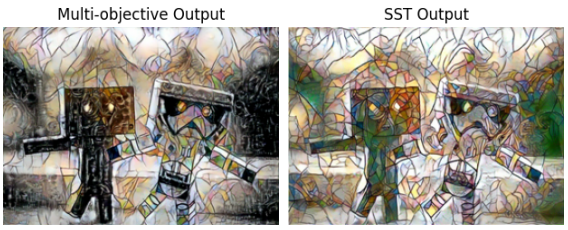

Figure 2: Outputs

ST-VAE is based on the COCO and WikiArt datasets for both single and multiple style transfer. Feed-forward training and testing is possible due to linear transformations making it very efficient. ST-VAE transfers textures and styles efficiently, it also successfully preserves the content information during style transition (Liu et al., 2021).

ViT enables perseverance of spatial information of the content structure and style patterns as it matches the style tokens onto content tokens in a fine-grained manner. ViT can serve to be computationally less expensive due to its iterative optimization process. Taking advantage of previous knowledge, this architecture could learn semantic-level correspondences between the content and style images and reduce distortion.

Optimizing an image is computationally expensive and contains no learned representations for the artistic style, which is inefficient for real-time changes.

No existing PyTorch based code implementation was found for the three evaluation metrics. All three metrics were implemented from scratch to incorporate into the existing code base for evaluation of results, of which two of the metrics need code revision as the values obtained are not interpretable based on the results. Besides, the current implementation is not efficient enough for one of the metrics and code could not be run on our local systems.

Based on our results, we plan on completing and refining our evaluation metric implementation for easier quantitative evaluation of our future experiments. This would enable faster iterations for our experiments and we will be able to start incorporating user feedback in a more concrete way.

References

Yingying Deng, Fan Tang, Xingjia Pan, Weiming Dong, ChongyangMa, and Changsheng Xu. 2021. Stytr^2 : $Unbiasedimagestyletransferwithtransformers$. *ArXiv*, $abs/2105.14576$.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *Journal of Vision*, abs/1508.06576.

Aleksa Gordić. 2020. Reference GitHub repository. `https://github.com/gordicaleksa/pytorch-neural-style-transfer`.

Zhi-Song Liu, Vicky Kalogeiton, and Marie-Paule Cani. 2021. Multiple style transfer via variational autoencoder. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2413–2417.

Jianbo Wang, Huan Yang, Jianlong Fu, T. Yamasaki, and Baining Guo. 2022. Fine-grained image style transfer with visual transformers. In *Asian Conference on Computer Vision*.

Zhizhong Wang, Lei Zhao, Haibo Chen, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. 2021. Evaluate and improve the quality of neural style transfer. *Computer Vision and Image Understanding*, 207:103203.

Zhizhong Wang, Lei Zhao, Qihang Mo, Sihuan Lin, Zhiwen Zuo, Wei Xing, and Dongming Lu. 2020. Cascade style transfer. *Under review as a conference paper at ICLR 2020*.

A Division of Labor

Each team member tried to divide all work equally. Attempts to understand and implement all 3 evaluation metrics were taken care of by Neha. Sneha worked on the implementation of the Vision Transformer based method for NST. Swarnita handled the implementation of the VAE based NST. Praveen worked on the implementation of the Gram matrix based method. All team members have collaborated smoothly using GitHub as the version control technology. All team members worked on contributing a set of references to the midterm report and the final addition to Overleaf Latex format.

B Appendix

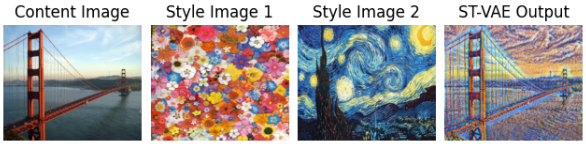

Figure 3: VAE Output