PARAMETER-FREE STYLE PROJECTION FOR ARBITRARY IMAGE STYLE TRANSFER

Siyu Huang[1*], *Haoyi Xiong*[2], *Tianyang Wang*[3], *Bihan Wen*[4],
Qingzhong Wang[2], *Zeyu Chen*[2], *Jun Huan*[5*], *Dejing Dou*[2]

[1]Harvard University, USA   [2]Baidu Research, China   [3]Austin Peay State University, USA
[4]Nanyang Technological University, Singapore   [5]Amazon, USA

ABSTRACT

Arbitrary image style transfer is a challenging task which aims to stylize a content image conditioned on arbitrary style images. In this task the feature-level content-style transformation plays a vital role for proper fusion of features. Existing feature transformation algorithms often suffer from loss of content or style details, non-natural stroke patterns, and unstable training. To mitigate these issues, this paper proposes a new feature-level style transformation technique, named Style Projection, for parameter-free, fast, and effective content-style transformation. This paper further presents a real-time feed-forward model to leverage Style Projection for arbitrary image style transfer, which includes a regularization term for matching the semantics between input contents and stylized outputs. Extensive qualitative analysis, quantitative evaluation, and user study have demonstrated the effectiveness and efficiency of the proposed methods.

Index Terms— Image style transfer, feature fusion, feature transformation, convolutional neural networks

1. INTRODUCTION

Arbitrary image style transfer [1] is a very challenging task which aims to synthesize artistic images conditioned on arbitrary styles, where the content-style feature transformation plays a vital role and a series of content-style transformation algorithms have been proposed in the literature [2–6]. The widely-used normalization-based methods, including instance normalization (IN) [7], conditional IN (CIN) [8], and adaptive IN (AdaIN) [9–11], generally normalize the style of content features to style features. However, only limited style information, *i.e.*, the first-order statistics of style features, is injected into the content features. Another widely-adopted feature-level style transformation method is whitening and coloring transformation (WCT) [12] which completely peels off the style information of content features then recovers with the target style information. Although the whitening operation was designed to remove style information, it generates artifacts that content details are also removed unintentionally, leading to unpleasant synthesized results. Both

normalization-based methods and WCT-based methods cannot well balance the content preserving and style transferring.

In this paper, we propose a new feature-level style transformation technique, named Style Projection, which is inspired by the order statistics [13, 14]. The order statistics-based image processing filters, such as max/min/median filters and ranked-order filters, can preserve fine content details of images in applications of noise detection [15, 16], image denoising [17], and object detection [18]. In this work, we exploit order statistics for deep neural network-based content-style feature transformation for the first time, by considering content/style information as order statistics/scalar values of features, respectively. In a very simple manner, Style Projection reorders the style features according to the order of the content features, such that the correlation of feature values (shapes, textures, etc.) is provided by the content features, while the color information is provided by the style features, enabling a reasonable content-style fusion.

We further present a learning-based feed-forward model to leverage Style Projection for arbitrary image style transfer in real time. The model works in an encoder-decoder fashion that stylizes a content image based on arbitrarily given style images. In experiments, we conduct extensive studies including quantitative evaluation, qualitative analysis, ablation studies, and user study, to comprehensively validate the efficacy of our style transfer framework. The contributions of this paper are summarized as follows: (1) We present a new parameter-free feature-level transformation technique, named Style Projection, for fast yet effective content-style feature fusion; (2) We present a real-time feed-forward method for arbitrary style transfer, including a KL divergence loss for further matching the semantics between input contents and stylized output; (3) We demonstrate the effectiveness and efficiency of our proposed methods through extensive empirical studies.

2. METHOD

2.1. Arbitrary Image Style Transfer

The goal of arbitrary image style transfer is to stylize a content image conditioned on an arbitrary style image. To

---

*Work done when Siyu Huang and Jun Huan were at Baidu Research.

Algorithm 1 Parameter-Free Style Projection.

Require: content feature map $x \in \mathbb{R}^{C \cdot H \cdot W}$
        style feature map $y \in \mathbb{R}^{C \cdot H \cdot W}$
Ensure: stylized feature map $z \in \mathbb{R}^{C \cdot H \cdot W}$

  1: $\vec{x} \in \mathbb{R}^{C \cdot V} \leftarrow$ reshape x;
      $\vec{y} \in \mathbb{R}^{C \cdot V} \leftarrow$ reshape y;
  2: index d_x, values $\vec{x}_r \leftarrow$ sort $\vec{x}$ along V;
      index d_y, values $\vec{y}_r \leftarrow$ sort $\vec{y}$ along V;
  3: $\vec{z}[:, i] \leftarrow \vec{y}_r[:, d_x[i]], \quad i = 1, 2, \ldots, V$;
  4: $z \in \mathbb{R}^{C \cdot H \cdot W} \leftarrow$ reshape $\vec{z}$.

---

address this task, we introduce a learning-based feed-forward style transfer model. Following the image style transfer practice in previous literature [9, 12], we embed content and style images into content and style features with an image encoder E. As discussed above, the critical component of our style transfer model is Style Projection algorithm, which is a parameter-free feature transformation approach for proper fusion of content and style features. Based on Style Projection, the content and style features are fused and then reconstructed as a new stylized image with an image decoder D. Despite the simplicity of the model, it is effective and there is a leeway to incorporate useful tricks, for instance the semantic matching regularization of KL divergence, to further boost the style transfer performance. In the following sections, we first introduce Style Projection algorithm and then discuss the learning objectives of our style transfer model.

2.2. Style Projection Algorithm

In this work, we introduce Style Projection which is a simple, fast, yet effective algorithm for content-style fusion. Given a content feature map $x \in \mathbb{R}^{C \cdot H \cdot W}$ and a style feature map $y \in \mathbb{R}^{C \cdot H \cdot W}$, where C, H, W are channel, height, and width respectively, we firstly vectorize x and y across dimension H and W to obtain features $\vec{x} \in \mathbb{R}^{C \cdot V}$ and $\vec{y} \in \mathbb{R}^{C \cdot V}$, where $V = H \cdot W$. Then we compute rankings for elements in $\vec{x}$ and $\vec{y}$. Afterwards, we reorder $\vec{y}$ by aligning each element to its corresponding same ranked element in $\vec{x}$. This actually reorganizes the style feature $\vec{y}$ according to the sorting order of the content feature $\vec{x}$. Then we reshape the adjusted feature to get $z \in \mathbb{R}^{C \cdot H \cdot W}$, which will be treated as the input of a decoder D to generate the stylized image. The computing process of Style Projection is shown in Alg. 1.

We propose that the Style Projection module will not lose the color details of style images, while the structure of content images are also carried in the transferred features. In the following, we take insight into the style and content preservation achieved by Style Projection algorithm, respectively.

Style detail preservation. To understand the effectiveness of Style Projectionon style detail preservation, we investigate the Gram matrix of images. The Gram matrix $G(i, j)$ of feature

Content / Style   No Shuffling   Random Shuffling   Style Projection

Fig. 1: An empirical study on feature shuffling mechanisms. "No Shuffling" directly feeds style features to the decoder without feeding content features or conducting any shuffling operation. "Random Shuffling" feeds randomly shuffled style features (shuffled within each channel of feature maps) to the decoder. "Style Projection" reorders style features according to order statistics of content features.

map y is formulated as

$$G(i, j) = \sum_p y(i, p) y(j, p) \qquad (1)$$

where i and j denote two channels on feature map y, and p denotes the spatial position. Such that $G_y(i, j)$ is the inner product between two feature channels.

The Gram matrix $G(i, j)$ can be used to evaluate texture synthesis algorithms by measuring the texture correlation between images [1]. Consider the difference between Gram matrices of feature maps y and z,

$$\begin{aligned} \mathcal{L}_{Gram}(y, z) &= ||G_y - G_z||_{\text{F}}^2 \\ &= \sum_{i,j} || \sum_p y(i, p) y(j, p) - \sum_p z(i, p) z(j, p) ||^2 \end{aligned}$$
$$(2)$$

Gu et al. [19] theoretically reveals that Eq. 2 is equal to 0 when y and z can be transformed to each other using a bijective transformation, for instance any shuffling function. Therefore, the feature shuffling does not alter the Gram matrix of feature. From Eq. 2, we show that the style information can be well preserved after style feature reshuffling in our proposed Style Projection module.

Content structure preservation. Style Projection also preserves the structure of content images. As suggested by [13], the order of random variables contains effective information where the order statistics are related to the distribution function of random variables. In Style Projection module, the style features are reshuffled according to the order of content features, thus the structural relationships in content features can be implicitly inherited by the reshuffled style features.

To verify this claim, we conduct an empirical study on feature shuffling mechanisms and the results are shown in Fig. 1. We observe that among the three methods, only Style Projection is capable of fusing content and style properly. 'No shuffling' of style features shows an exact reconstruction of the original style image. 'Random shuffling' shows a repetition of random style patterns, demonstrating that the style patterns can be preserved after feature reshuffling. Only Style
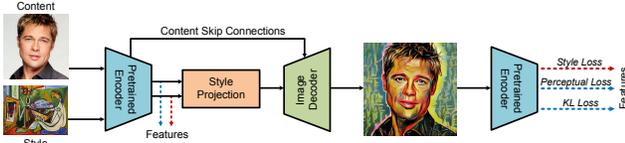
Fig. 2: The scheme of our proposed style transfer framework.

Projection, which reorders the style features based on content features, shows reasonable image stylization. This study reveals the effectiveness of Style Projection as it is able to preserve both content and style information during feature transformation. Through Style Projection, the spatial structure (shape, edge, etc.) of content images and the color/texture patterns of style images are well fused.

2.3. Style Transfer Model

We illustrate our style transfer model in Fig. 2. The learning objective of our style transfer model is composed of three parts, including the style loss, the content perceptual loss, and the content KL divergence loss. The style loss $\mathcal{L}_s$ is used to match the feature statistics between the style image s and the stylized image $\hat{c}$ as $\mathcal{L}_s = \sum_{i=1}^{N} \|\mu, \sigma(E_i(s)) - \mu, \sigma(E_i(\hat{c}))\|_2$, where μ and σ denotes the mean and standard deviation, and E_i is the intermediate output of the i-th layer of encoder E, and N is the number of encoder layers. The content perceptual loss $\mathcal{L}_p$ [20] is used to minimize the pixel-wise feature distance between the content image c and the stylized image $\hat{c}$ as $\mathcal{L}_p = \|E(c) - E(\hat{c})\|_2$.

Most image style transfer methods suffer from non-natural image clues in stylized results. We conjecture this is partially due to the missing of semantic information such as brightness. Inspired by the insights provided by generative models [21], we introduce a distribution matching objective, Kullback-Leibler (KL) divergence [22], into the style transfer framework to leverage more semantic information in content images, as

$$\mathcal{L}_{KL} = \mathcal{KL}\left[E(c) \,||\, E(\hat{c})\right] \qquad (3)$$

where c is an input content image and $\hat{c}$ is the stylized image produced by decoder D. With the aid of KL divergence, we regularize D to generate images that contain more semantics of content images.

Overall, the complete learning objective of our model is formulated as $\mathcal{L} = \mathcal{L}_p + \lambda \mathcal{L}_s + \kappa \mathcal{L}_{KL}$, where λ and κ are the loss weights of $\mathcal{L}_s$ and $\mathcal{L}_{KL}$, respectively. As Style Projection is a parameter-free method and the added KL divergence loss only brings negligible extra computing time, our style transfer approach is highly efficient.

Table 1: Quantitative evaluation of style loss $\mathcal{L}_s$, content loss $\mathcal{L}_p$, and total loss. The lower the better.

Method	Style loss $\mathcal{L}_s$	Content loss $\mathcal{L}_p$	Total
CNN [1]	0.90	2.51	3.41
StyleSwap [29]	4.87	2.56	7.43
WCT [12]	1.33	4.08	5.41
AdaIN [9]	0.39	2.56	2.95
LinearPropagation [30]	2.86	2.86	5.73
Style Projection	**0.38**	2.61	2.99
Style Projection + Skip	0.48	2.56	3.04
Style Projection + KL	**0.38**	2.25	2.63
Style Projection + Skip + KL	0.49	**2.09**	**2.58**

3. EXPERIMENTS

3.1. Experimental Setup

Datasets. We adopt the training set of MS-COCO dataset [23] as the content images and that of WikiArt dataset [24] as the style images. In training, we resize all input images to the size of 512×512 and randomly crop each image to 256×256. All content and style images used for testing purpose are selected from the test set of the two datasets, and the test images are never observed by the model during training. Our encoder and decoder networks work in a fully-convolutional manner, and they can be applied to images of arbitrary size in testing.

Implementation. We implement our style transfer model[1] based on the PaddlePaddle framework [25] and release a pre-trained model as an official PaddleHub tool[2]. We train our style transfer model for 160,000 iterations. We use an Adam optimizer [26] with an initial learning rate of 1e-4 and a learning rate decay of 5e-5. Unless otherwise specified, the loss weights λ and κ are set to 10 and 2.5, respectively. Note that the loss weights are employed to balance the style transfer and content semantics preservation. More empirical studies on the KL divergence weight κ are illustrated in the following ablation studies. We use a batch size of 8 for training.

Networks. The encoder network used in the model is the VGG-19 network [27] pre-trained on ImageNet [28], and its weights are not updated during training. The decoder network is composed of nine 'Padding-Conv-ReLU' blocks, except the last block that has no ReLU layer. Three up-sampling layers are adopted right after the 1-st, 5-th, and 7-th block to restore the input image dimension successively, where the nearest neighbor interpolation is employed for up-sampling. We do not use normalization layer in our decoder network since it will hurt the diversity of synthesized images [9].

3.2. Comparison with State-of-the-Arts

Quantitative evaluation. In Fig. 3, we quantitatively evaluate the state-of-the-art arbitrary image style transfer methods, including Style-Swap [29], WCT [12], AdaIN [9], Linear Propagation [30]. We investigate the content loss $\mathcal{L}_p$, style

---

[1]https://github.com/PaddlePaddle/PaddleHub/Stylepro_Artistic
[2]https://www.paddlepaddle.org.cn/hubdetail?name=stylepro_artistic

2072

Fig. 3: Qualitative comparisons between the state-of-the-art content-style transformation modules. Ours-2 adds content skip connections and KL divergence loss to our Style Projection algorithm (Ours-1) to deliver more appealing images.

Fig. 4: A comparison of shuffling-based transfer methods.

Table 2: User study on style transfer methods. "**Method Score**" is the percentage of times that users prefer the images in pairwise comparisons. "SP" denotes Style Projection and "Full" denotes Style Projection+Skip+KL. "**User Consistency**" denotes the consistency among users. "Scene" and "Person" are two categories of content images.

	Method Score (%)			User Consistency (%)
	AdaIN	SP	Full	
Scene	32.58	51.89	**65.53**	68.94
Person	34.47	37.50	**78.03**	76.52

loss $\mathcal{L}_s$ and total loss in inference. Our Style Projection shows competitive results compared with the state-of-the-art style transfer modules with respect to both style and content loss. The SP+Skip+KL method presents more superior results than the other methods and similar style loss as AdaIN, indicating that the modules proposed in this paper, including Style Projection and KL divergence loss, work well in arbitrary style transfer tasks. We also observe that there is a trade-off between style loss and content loss. Generally, SP+KL and SP+Skip+KL reach a good balance in such a trade-off.

Qualitative comparison. In Fig. 3 we show stylized images generated by different methods. The images generated by WCT [12] have a low fidelity probably due to the whitening operation which peels off some critical content clues. AdaIN [9] better reconstructs the content details, however there are still several 'non-natural strokes' caused by only transferring the mean and standard deviation of the style features. The images produced by the StyleSwap [29] are too dark probably due to the way of replacing content features. Our methods (last two columns) provide more pleasant results. Even using Style Projection independently (*i.e.*, Ours-1), style information is properly transferred while main content is preserved. SP+Skip+KL (*i.e.*, Ours-2) produces more appealing results where content details and semantics are better preserved.

Both DFR [19] and SP are reshuffling-based feature transformation methods. A comparison between DFR and Style Projection is shown in Fig. 4. DFR includes an expensive optimization process in its framework, thus it is time-consuming in image generation (taken about 114 seconds/image). Our parameter-free Style Projection is able to give appealing stylized results with a high efficiency (0.068 second/image).

3.3. User Study

We further conduct a user study to evaluate three different methods, including AdaIN, our SP, and our full model (SP+Skip+KL). We build 24 pairs of synthesized images including 8 pairs of AdaIN and SP, 8 pairs of AdaIN and Full, and 8 pairs of SP and Full. The users are asked to choose one synthesized image from each pair under the criteria: (1) whether the image is sharp and clean while its content is close to that of the content image, and (2) whether the image has a similar style as the style image. We have collected feedbacks from 33 individuals and the results are summarized in Table 2. The 'Method Score' is the preference ratio averaged over all the image pairs and users, and the results indicate that most users favor the synthesized images of our methods. Interestingly, for the person images, more users prefer our full model (78.03%), which indicates that fidelity is deserved to be well maintained for person images (see Fig. 3).

4. CONCLUSION

In this paper, we have presented a real-time feed-forward model for arbitrary style transfer. The core is a parameter-free, fast, and effective feature-level style transformation algorithm named Style Projection. We have also introduced the KL divergence loss into our style transfer model for a regularization of semantic consistency on content structures. Extensive experiments including quantitative evaluation, qualitative analysis, and user study have validated the efficacy of our method for arbitrary image style transfer.

5. REFERENCES

[1] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in *CVPR*, 2016, pp. 2414–2423. 1, 2, 3

[2] David J Heeger and James R Bergen, "Pyramid-based texture analysis/synthesis," in *Annual conference on computer graphics and interactive techniques*, 1995, pp. 229–238. 1

[3] B. Zhao, H. Xiong, J. Bian, Z. Guo, C. Xu, and D. Dou, "Como: Widening deep neural networks with convolutional maxout," *IEEE Transactions on Multimedia*, pp. 1–1, 2020. 1

[4] Jeremy S De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Annual conference on computer graphics and interactive techniques*, 1997, pp. 361–368. 1

[5] Javier Portilla and Eero P Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *IJCV*, vol. 40, no. 1, pp. 49–70, 2000. 1

[6] Zhi-Qi Cheng, Xiao Wu, Yang Liu, and Xian-Sheng Hua, "Video ecommerce++: Toward large scale online video advertising," *IEEE transactions on multimedia*, vol. 19, no. 6, pp. 1170–1183, 2017. 1

[7] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *CVPR*, 2017, pp. 6924–6932. 1

[8] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016. 1

[9] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *CVPR*, 2017, pp. 1501–1510. 1, 2, 3, 4

[10] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019, pp. 2337–2346. 1

[11] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410. 1

[12] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang, "Universal style transfer via feature transforms," in *NIPS*, 2017, pp. 386–396. 1, 2, 3, 4

[13] Herbert Aron David and Haikady Navada Nagaraja, "Order statistics," *Encyclopedia of Statistical Sciences*, 2004. 1, 2

[14] Ioannis Pitas and Anastasios N Venetsanopoulos, "Order statistics in digital image processing," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1893–1921, 1992. 1

[15] Nemanja I Petrovic and Vladimir Crnojevic, "Universal impulse noise filter based on genetic programming," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1109–1120, 2008. 1

[16] Pei-Eng Ng and Kai-Kuang Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1506–1516, 2006. 1

[17] Henry Pinkard, Kaitlin Corbin, and Matthew F Krummel, "Spatiotemporal rank filtering improves image quality compared to frame averaging in 2-photon laser scanning microscopy," *PloS one*, vol. 11, no. 3, pp. e0150430, 2016. 1

[18] Atindra K Mitra, Thomas L Lewis, and Arnab K Shaw, "Rank-order filters for fopen target detection," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 93–96, 2004. 1

[19] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan, "Arbitrary style transfer with deep feature reshuffle," in *CVPR*, 2018, pp. 8222–8231. 2, 4

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016, pp. 694–711. 3

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680. 3

[22] Solomon Kullback and Richard A Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951. 3

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755. 3

[24] K Nichol, "Painter by numbers, wikiart," https://www.kaggle.com/c/painter-by-numbers, 2016. 3

[25] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang, "Paddlepaddle: An open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105–115, 2019. 3

[26] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 3

[27] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 3

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255. 3

[29] Tian Qi Chen and Mark Schmidt, "Fast patch-based style transfer of arbitrary style," *arXiv preprint arXiv:1612.04337*, 2016. 3, 4

[30] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang, "Learning linear transformations for fast image and video style transfer," in *CVPR*, 2019, pp. 3809–3817. 3