

SAFIN: ARBITRARY STYLE TRANSFER WITH SELF-ATTENTIVE FACTORIZED INSTANCE NORMALIZATION

Aaditya Singh^{*†1}, Shreeshail Hingane^{*1}, Xinyu Gong², Zhangyang Wang²

¹ Indian Institute of Technology Kanpur, ² The University of Texas at Austin

ABSTRACT

Artistic style transfer aims to transfer the style characteristics of one image onto another image while retaining its content. Existing approaches commonly leverage various normalization techniques, although these face limitations in adequately transferring diverse textures to different spatial locations. Self-Attention based approaches have tackled this issue with partial success but suffer from unwanted artifacts. Motivated by these observations, this paper aims to combine the best of both worlds: self-attention and normalization. That yields a new plug-and-play module that we name *Self-Attentive Factorized Instance Normalization (SAFIN)*. SAFIN is essentially a spatially-adaptive normalization module whose parameters are inferred through attention on the content and style image. We demonstrate that plugging SAFIN into the base network of another state-of-the-art method results in enhanced stylization. We also develop a novel base network composed of Wavelet Transform for multi-scale style transfer, which when combined with SAFIN, produces visually appealing results with lesser unwanted textures.

1. INTRODUCTION

Artistic style transfer is the task of extracting style and texture patterns from one image and transferring them onto the content of another. Initial work by [3] employed an optimization based approach. This approach was replaced by feed-forward networks that could generate the images in a single forward pass [4] [5], with the major limitation being that these had to be retrained for every new style image. Further work on doing *multiple-styles-per-model* followed [6], though these could not effectively generalize to new unseen styles.

Further research led to models that perform style transfer for *arbitrary* styles [2] [7]. Adaptive Instance Normalization (AdaIN) [2] was one of the enabling techniques, which adjusted the mean and variance of the content image feature-maps to match those of the style image. The transferred results, though often decent, also appear to be over-simplistic and lack fine details, since only simple statistic moments were drawn for content-style matching. Following this idea, many



Fig. 1: Stylization results. A few stylization results generated using SAFIN with our base network.

other normalization based approaches such as Batch-IN [9] and Dynamic-IN [8] were proposed.

Despite much progress being made on normalization-based techniques, one remaining challenge lies in the lack of ability to transfer different textures to different spatial locations of the image with semantic awareness. Although spatially adaptive normalization has been recently used to good effect in other tasks [17] [1], similar ideas have not been explored in style transfer yet.

Attention-based models [10] [11] have shown another successful stream in style transfer. They attempt to resolve this limitation by employing self-attention over the spatial dimensions when transforming the content features. However, these sometimes cause unwanted patterns in output (Fig. 3).

Motivated by these observations, this paper aims to combine the best of both worlds: self-attention and normalization. Our underlying motivation is to make normalization more spatially flexible and semantically aware. We achieve this by using a self-attention based module to learn to gen-

* Equal contribution

† Currently affiliated with IBM Research

erate the parameters of our spatially-adaptive normalization module. We further introduce a new factorized parameter structure to the normalization layer. The resulting new module, called Self-Attentive Factorized Instance Normalization (SAFIN), has a plug-and-play nature and can boost other state-of-the-art normalization-based style transfer methods. In addition, we develop a novel base network composed of Wavelet Transform for multi-scale style transfer. When combined with SAFIN, visually appealing stylization results can be obtained without additional artifacts and distortions.

2. RELATED WORK

2.1. Normalization in NST

The essential task in Neural Style Transfer (NST) is the transformation of content-image features in a way that their style characteristics match those of the style image.

In optimization based methods (i.e. *single-style-per-model*) [5] [4], the use of batch-normalization (BN) [13] was purely for its standard purpose of easing neural network training. The work by [14] introduced instance-normalization (IN) and its advantage over batch normalization due to its ability to discard instance-specific contrast information. With new models performing *multiple-styles-per-model* or *arbitrary-styles-per-model*, normalization (i.e affine shifting and scaling of mean-variance-normalized content features) became a way of imbuing the content features with the style characteristics of the input style. For instance, distinct scaling/shifting parameters are learned for each new style during training in Conditional-IN [6], thus allowing the model to achieve stylization for multiple styles.

The main task in accomplishing *arbitrary* style transfer using the normalization based approach is to compute the normalization parameters at test time. AdaIN [2] showed that even parameters as simple as the channel-wise mean and variance of the style-image features could be effective. Batch-IN [9] used a combination of BN and IN whereas Dynamic-IN [8] used a learned convolutional network to extract the scaling and shifting parameters given the style and content images.

A major limitation of the current normalization based approaches is their inability to transfer diverse and fine stroke patterns from different parts of the style image to the semantically similar regions in the content image (Fig 2). This is because the normalization is either not spatially adaptive [2] [9] or lacks an appropriate mechanism to infer and transfer the diversity in style patterns onto the normalization parameters.

2.2. Other Techniques

WCT [7] performs the transformation on the content features by using the whitening and coloring transform. Not only does WCT suffer from transfer of global style characteristics, leading to less varied stroke patterns, but it is also computation-

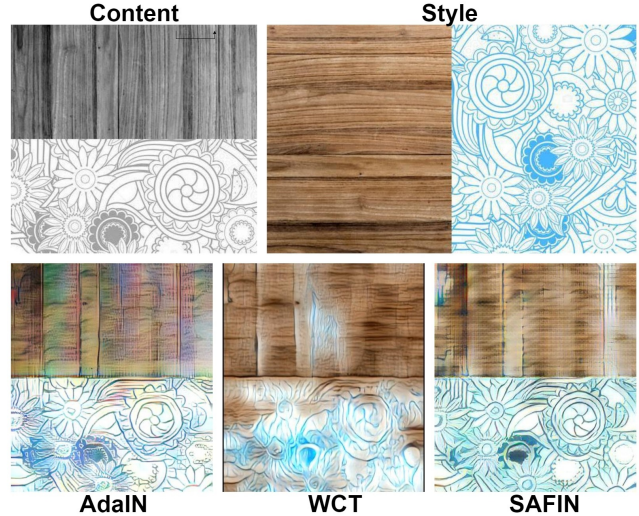


Fig. 2: Semantic awareness. Attention-based models show some capability of matching patterns to decide what texture to transfer to different spatial locations. SAFIN has no obvious spillover, whereas WCT and AdaIN suffer from spillovers.

ally expensive and especially slow and memory consuming for high-resolution images.

Initial attempts to solve the problem of transferring different style patterns used a patch-based approach by matching local statistics between semantically matching patches [18] [19]. Though these methods achieve visually pleasing results, their application is limited to content-style combinations with dense semantic correspondence.

2.3. Attention in NST

Self-Attention [20] has been applied to the problem of style transfer in SANet [10] since it suggests a potential way to designate different style patterns to different regions in an image. However, in SANet, the fine-grained nature of the attention target and direct incorporation within the content feature maps without further processing sometimes leads to extraneous artifacts and distortions (Fig. 3).

Our method builds upon the ability of SANet and attempts to avert its drawbacks by using it in a novel manner. We use the self-attention module as a part of a convolutional network trained to infer the normalization parameters. This is in contrast to SANet, which directly uses the attention output to infer the feature map for the stylized output.

3. APPROACH

3.1. Factorized Instance Normalization (FIN)

We present a novel normalization module called Factorized Instance Normalization (FIN). Given the content feature maps $F_c \in \mathbb{R}^{H*W*C}$ we transform the style characteristics of these

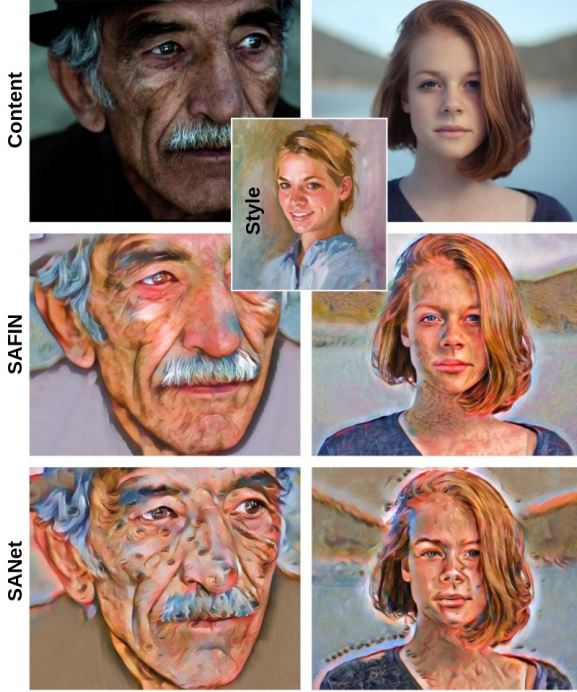


Fig. 3: Extraneous artifacts. SANet produces eye-like artifacts (zoom in for greater clarity) due to the fine-grained nature of its attention target while SAFIN does not.

features to match the style characteristics of the style image while maintaining its semantic information. We achieve this via the following transformation:

$$\text{FIN}(F_c) = \gamma_s \odot (\gamma_{ind} \odot \overline{F_c} + \beta_{ind}) + \beta_s \quad (1)$$

where $\overline{F_c} \in \mathbb{R}^{H*W*C}$ is the normalized content feature obtained by applying channel-wise instance normalization on F_c . \odot denotes element-wise multiplication.

For the shifting and scaling operations (Eq. 1), we use two types of learned normalization parameters:

1. γ_{ind} and β_{ind} represent the *style independent* parameters and are single dimension vectors along the channel dimension (i.e. $\gamma_{ind}, \beta_{ind} \in \mathbb{R}^C$). These parameters are meant to capture the shared transformation between the different style images, by transforming $\overline{F_c}$ before the style dependent transformation.
2. On the other hand, γ_s and β_s are inferred from the input style image. γ_s, β_s are 3 dimensional; along the channel dimension as well as the spatial dimensions (i.e. $\gamma_s, \beta_s \in \mathbb{R}^{H*W*C}$). Thus, element-wise shifting and scaling of the content features using γ_s and β_s allows for greater flexibility in transferring diverse textures to different spatial regions on the content image. However, to generate rich stylization, it is not enough to simply use spatially adaptive normalization, since a mechanism is required that can adequately learn to

derive these normalization-parameters from the input style image. Thus, γ_s and β_s are inferred from the style image using a network employing self-attention followed by convolution. This mechanism is explained in subsection 3.2

3.2. Self-Attention Mechanism (SA)

We employ Self-Attention [20] to capture the semantic correspondence between the content feature maps F_c and the style feature maps F_s (extracted from the encoder) for generating the style-based normalization parameters γ_s and β_s .

$$\gamma_s = \text{RELU}(W_\gamma \otimes \text{SA}(\overline{F_c}, \overline{F_s})) \quad (2)$$

$$\beta_s = \text{RELU}(W_\beta \otimes \text{SA}(\overline{F_c}, \overline{F_s})) \quad (3)$$

Here $\overline{F_c}, \overline{F_s} \in \mathbb{R}^{H*W*C}$ are channel-wise instance normalized content and style feature maps respectively. W_γ, W_β represent $1 * 1$ convolutional layers and \otimes denotes the convolution operation. Our Self-Attention module (SA) is similar to that used by [10], though it is employed for a different purpose. [10] used self-attention to directly modify the content feature maps, whereas we instead use it to generate the style-dependent normalization parameters of our FIN module.

3.3. Wavelet Transform

For better preservation of content features in the stylized outputs, we use Wavelet Pooling (to replace Max Pooling) and Wavelet Unpooling (to replace Upsampling) [24], a pooling technique which allows exact reconstruction due to its invertible nature. This technique has been previously employed in photo-realistic style transfer [12], but not, to the best of our knowledge, in artistic style transfer.

Wavelet pooling uses four distinct pooling kernels $\{LL^\top, LH^\top, HL^\top, HH^\top\}$ at every pooling layer, made by combining the high-pass filter $H^\top = \frac{1}{\sqrt{2}}[-1 \ 1]$ and the low-pass filter $L^\top = \frac{1}{\sqrt{2}}[1 \ 1]$. We use the terms LL, LH, HL, and HH to refer to the output of the respective kernel.

3.4. Network Architecture

Figure 4 shows the network architecture, which follows the commonly employed encoder-decoder structure [2], [7], [10]. The encoder has the same structure as the VGG-19 network, and the decoder is symmetric to the encoder.

The loss function for training the SAFIN module and the decoder is given as: $L = L_c + \lambda_s * L_s$, where L_c and L_s are content and style losses and λ_s is the weight assigned to the style loss relative to the content loss.

The content loss is the Euclidean distance between LL components of stylised output image encoding $\mathcal{E}(I_{cs})$ and content image encoding $\mathcal{E}(I_c)$ extracted from the layer ReLU_4-1: $L_c = \|\mathcal{E}_{LL}^{4-1}(I_{cs}) - \mathcal{E}_{LL}^{4-1}(I_c)\|_2$

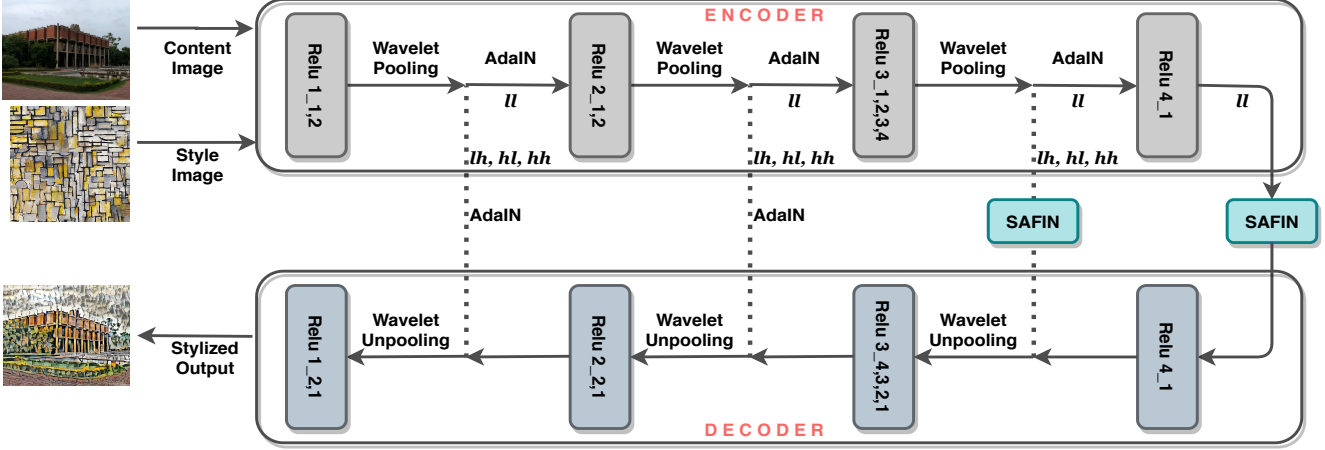


Fig. 4: Network Architecture. We train a multi-scale encoder-decoder network with two SAFIN modules and Wavelet Transform to stylize the low frequency component (LL, shown through bold lines) as well as the high frequency components (LH, HL, and HH, shown through dotted lines). AdaIN [2] is used at different scales to further enrich the stylization.

The style loss is the Euclidean distance between mean and variance of LL components of stylised output image encoding $\mathcal{E}(I_{cs})$ and style image encoding $\mathcal{E}(I_s)$ from the layers ReLU_X_1 where $X \in \{1, 2, 3, 4\}$:

$$L_s = \sum_{k=1}^4 \left(\|\mu(\mathcal{E}_{LL}^{k,1}(I_{cs})) - \mu(\mathcal{E}_{LL}^{k,1}(I_c))\|_2 + \|\sigma(\mathcal{E}_{LL}^{k,1}(I_{cs})) - \sigma(\mathcal{E}_{LL}^{k,1}(I_c))\|_2 \right)$$

Further details of the network architecture are described in the appendix.

4. EXPERIMENTS

We trained our network using MS-COCO dataset [15] for content images and WikiArt dataset [16] for style images. The detailed experimental settings are included in the appendix. Code and pretrained model are available here <https://github.com/Aaditya-Singh/SAFIN>.

4.1. Generalizability

To demonstrate the superiority as well as flexibility of our spatially adaptive normalization technique, we show how SAFIN module when substituted in to replace the SANet [10] module within the SANet base network, also mitigates the extraneous artifacts to some extent (Fig. 5a). The training-loss comparison plot (Fig. 5b) shows a lower content loss with the SAFIN module compared to the SANet module thus demonstrating that the usage of SAFIN results in better content preservation. Style losses are similar. Indeed, our module when inserted into SANet’s base network provides a comparable quality of stylization along with reduced unwanted distortions.

4.2. Ablation Study

4.2.1. Effect of removal of Self-Attention from SAFIN

In order to understand the effectiveness of Self-Attention (SA) mechanism within the overall SAFIN module, we compare the results when SAFIN module is replaced with FIN module (normalization without self-attention) in our network. From figure 6, it can be observed that while FIN module enables the transfer of general color and texture from the style image, self-attention allows the network to transfer the style characteristics from semantically similar regions in the style image leading to an enhanced stylization.

Further ablation studies and more stylization outputs for comparison are included in the appendix.

4.3. Quantitative Analysis

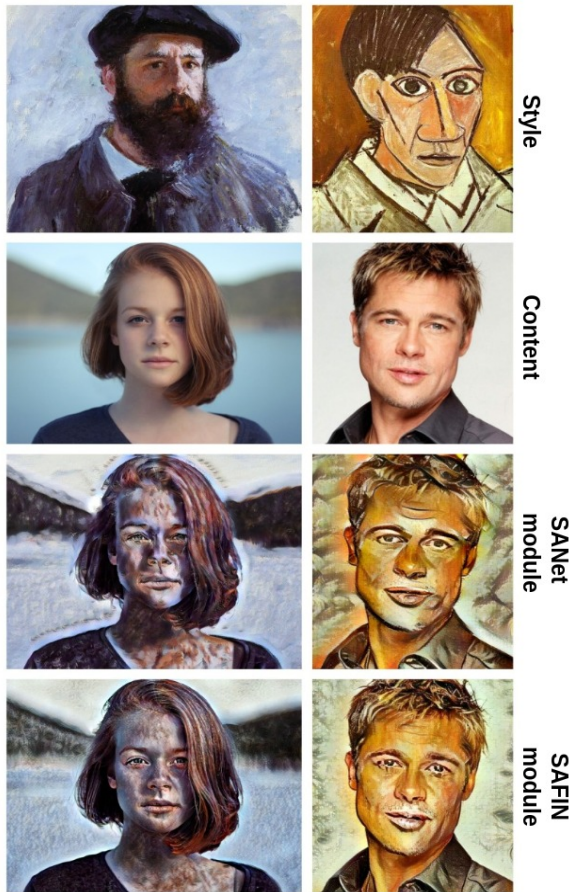
We also conducted an extensive survey for a thorough quantitative analysis. There were 25 participants and 40 sets of content-style image pairs along with their stylization outputs using 4 different methods - SAFIN (with the base network), SANet, WCT and AdaIN. For every set, each participant was asked two questions:

- To choose the best stylization, i.e. which method leads to the best color and texture transfer.
- To choose which stylization caused the least amount of unwanted artifacts and distortions.

The results (Fig. 7) show that our method is preferred by users against other methods, providing better or comparable stylization without leading to additional distortions.

5. CONCLUSION

In this paper, we propose a novel module called SAFIN which combines normalization and self-attention in a unique



(a) SAFIN module when substituted in place of the SANet module in SANet’s base network, leads to reduced unwanted distortions.



(b) Training loss comparison: SANet’s base network with SANet’s style transfer module vs SAFIN module.

Fig. 5: SAFIN (with SANet’s base network) vs SANet.



Fig. 6: Effect of removal of Self-Attention (SA) module. FIN transfers the general color and texture patterns but causes spillovers, whereas SAFIN produces a more semantically coherent stylization.

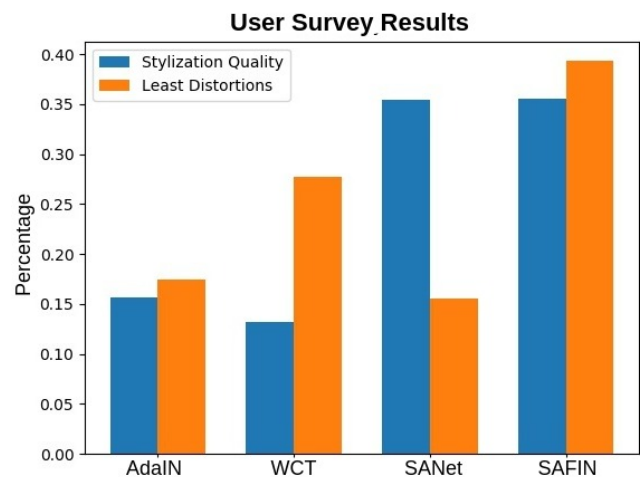


Fig. 7: User Survey Results. SAFIN with our base network causes the least amount of artifacts and distortions without compromising the stylization quality.

way to mitigate the issues with previous style transfer techniques. Extensive experiments demonstrate its effectiveness. We show the generalization capability of the proposed module by showing how it improves the stylization and losses of another state-of-the-art method when incorporated within its base network. We also develop a novel base network composed of Wavelet Transform for multi-scale style transfer that helps in preserving more content information. Experimental results illustrate the effectiveness of the proposed method.

6. REFERENCES

- [1] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.
- [2] Xun Huang and Serge Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.
- [3] 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*, 2016, pp. 2414–2423.
- [4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European conference on computer vision*. Springer, 2016, pp. 694–711.
- [5] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky, “Texture networks: Feed-forward synthesis of textures and stylized images.,” in *ICML*, 2016, vol. 1, p. 4.
- [6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur, “A learned representation for artistic style,” *arXiv preprint arXiv:1610.07629*, 2016.
- [7] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang, “Universal style transfer via feature transforms,” in *Advances in neural information processing systems*, 2017, pp. 386–396.
- [8] Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen, “Dynamic instance normalization for arbitrary style transfer,” *arXiv preprint arXiv:1911.06953*, 2019.
- [9] Hyeonseob Nam and Hyo-Eun Kim, “Batch-instance normalization for adaptively style-invariant neural networks,” in *Advances in Neural Information Processing Systems*, 2018, pp. 2558–2567.
- [10] Dae Young Park and Kwang Hee Lee, “Arbitrary style transfer with style-attentional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5880–5888.
- [11] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang, “Attention-aware multi-stroke style transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1467–1475.
- [12] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha, “Photorealistic style transfer via wavelet transforms,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9036–9045.
- [13] Sergey Ioffe and Christian Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [14] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.
- [15] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [16] Fred Phillips and Brandy Mackintosh, “Wiki art gallery, inc.: A case for critical thinking,” *Issues in Accounting Education*, vol. 26, no. 3, pp. 593–608, 2011.
- [17] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang, “Neural pose transfer by spatially adaptive instance normalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5831–5839.
- [18] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand, “Style transfer for headshot portraits,” 2014.
- [19] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang, “Visual attribute transfer through deep image analogy,” *arXiv preprint arXiv:1705.01088*, 2017.
- [20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [21] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*, 2015, pp. 2048–2057.
- [23] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola, “Stacked attention networks for image question answering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 21–29.
- [24] Jin-Sheng Guf and Wei-Sun Jiang, “The haar wavelets operational matrix of integration,” *International Journal of Systems Science*, vol. 27, no. 7, pp. 623–628, 1996.

SAFIN: ARBITRARY STYLE TRANSFER WITH SELF-ATTENTIVE FACTORIZED INSTANCE NORMALIZATION

TECHNICAL APPENDIX

Aaditya Singh^{*†1}, Shreeshail Hingane^{*1}, Xinyu Gong², Zhangyang Wang²

¹ Indian Institute of Technology Kanpur, ² The University of Texas at Austin

1. MOTIVATION

To elaborate more on the motivation of combining self-attention and spatially adaptive normalization, we present the specific drawbacks with current techniques.



Fig. 1: Fine textures. Using spatially adaptive normalization (in SAFIN) can help capture fine-textures and provide a more vivid stylization compared to AdaIN.

Having spatially adaptive normalization provides greater capability of capturing and adapting fine textures thus leading to richer and more vivid stylization when compared to normalization methods that are not spatially adaptive (Fig. 1).

With SAFIN, we also attempt to bridge the gap towards semantically-aware style transfer. This is not possible in AdaIN due to its lack of spatial adaptivity. This is also not possible in methods like WCT that perform style transfer using global characteristics. To demonstrate the potential of attention-based models, we present a simplified case (Fig. 2) in which we see how the lack of semantic awareness when using techniques which perform global style transfer leads to spillover effects.

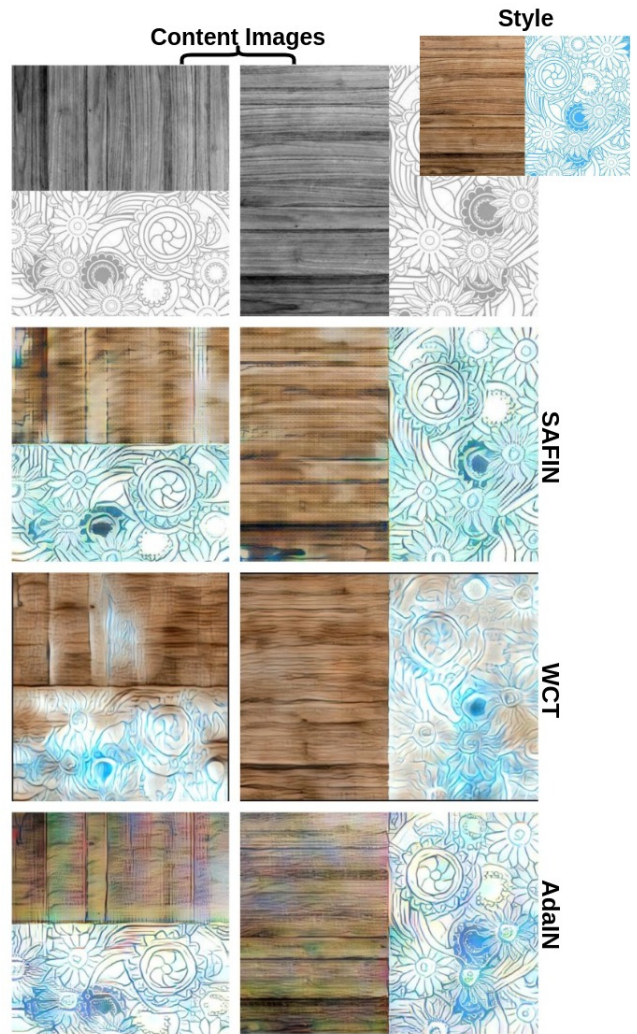


Fig. 2: Semantic awareness. Attention-based models show some capability of matching patterns to decide what texture to transfer to different spatial locations on the image. SAFIN has no obvious spillover, unlike WCT and AdaIN that suffer from spillovers

^{*} Equal contribution

[†] Currently affiliated with IBM Research

SANet [1] attempts to resolve the problem of content distortion using additional loss terms that require careful hyper-

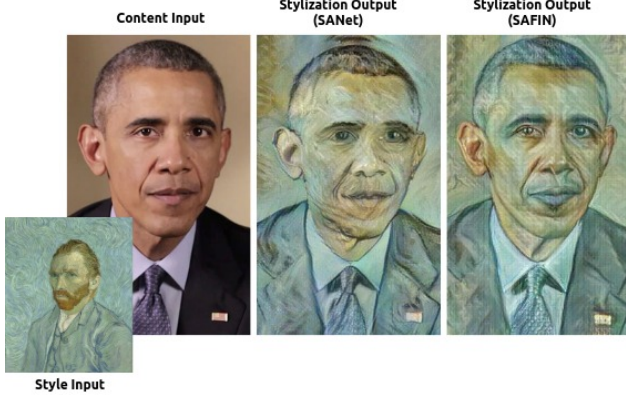


Fig. 3: Slight distortion of facial features in SANet. Even small distortions could potentially lead to a different meaning. Here, the positioning of the eyes changes the direction in which the person is looking.

parameter selection. Inspired by the work of [2], we approach this problem instead by using Wavelet Pooling and Unpooling to replace the Max Pooling and Upsampling operations, since the Haar Wavelet Transform kernels can help achieve better reconstruction on the semantic level. Figure 3 shows how SANet might cause image distortion. Sometimes, even small distortions can alter the meaning of a photograph. Therefore, faithful image reconstruction becomes a desirable property of style transfer methods.

Perhaps more importantly, SANet suffers from extraneous artifacts and distortions due to the the fine-grained nature of its attention target and direct incorporation within the content feature maps without further processing (Fig. 4). On the other hand, we use the self-attention as a part of a convolutional network trained to infer the normalization parameters.

2. NETWORK ARCHITECTURE DETAILS

2.1. SAFIN module

Figure 5 describes additional details of the proposed SAFIN module. We employ self-attention mechanism [3], [4] to capture the semantic correspondence between the content feature maps F_c and the style feature maps F_s extracted from the encoder for generating the style-based re-normalization parameters γ_s and β_s . The Self-Attention (SA) Module is similar to that used by [1]. \bar{F}_c and \bar{F}_s are the channel-wise mean-variance normalized content and style feature maps respectively. Attention between \bar{F}_c^i and \bar{F}_s^j is calculated as follows:

$$\text{SA}(\bar{F}_c^i, \bar{F}_s^j) = \frac{\sum_j \exp(f(\bar{F}_c^i)^T g(\bar{F}_s^j)) h(\bar{F}_s^j)}{\sum_j \exp(f(\bar{F}_c^i)^T g(\bar{F}_s^j))} \quad (1)$$

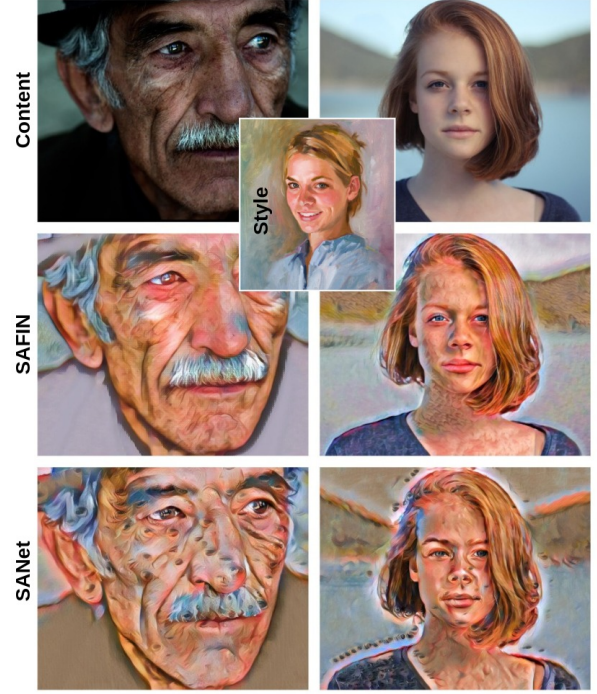


Fig. 4: Extraneous artifacts. SANet produces eye-like artifacts (zoom in for greater clarity) due to the fine-grained nature of its attention target while SAFIN does not.

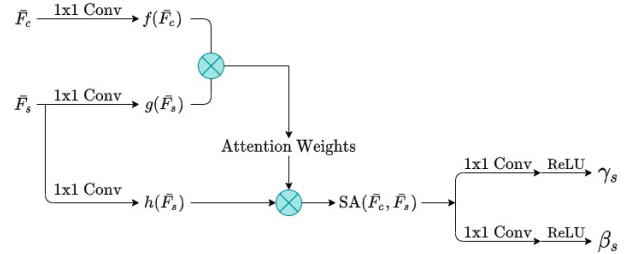


Fig. 5: SAFIN Module.

where $f(\bar{F}_c) = W_f \otimes \bar{F}_c$, $g(\bar{F}_s) = W_g \otimes \bar{F}_s$, and $h(\bar{F}_s) = W_h \otimes \bar{F}_s$ and W_f , W_g , and W_h are learned weight matrices that are implemented as 1×1 convolutions. Here, i is the output position index and the index j enumerates over all possible positions. We then obtain γ_s and β_s from $\text{SA}(\bar{F}_c, \bar{F}_s)$ by using a single layer convolutional network:

$$\gamma_s = \text{RELU}(W_\gamma \otimes \text{SA}(\bar{F}_c, \bar{F}_s)) \quad (2)$$

$$\beta_s = \text{RELU}(W_\beta \otimes \text{SA}(\bar{F}_c, \bar{F}_s)) \quad (3)$$

W_γ , W_β represent $1 * 1$ convolutional layers and \otimes denotes the convolution operation.

2.2. Network Architecture

This subsection describes additional details about the base network architecture (Fig. 4 in the main paper).

We apply SAFIN module to the low frequency LL component (of the image encoding output of the layer ReLU_4_1) for capturing smooth surface and texture information and the high frequency components LH, HL, and HH (encoding outputs of the layer ReLU_3_4) to stylize edges. For an enhanced low-level stylization (eg: colors) and high-level stylization (eg: contours) without compromising the inference speed, we replace the Whitening and Coloring Transform (WCT) [2] used in the lower layers of the encoder with Adaptive Instance Normalization (AdaIN) [5] module.

3. ABLATION STUDY

3.1. Experimental Settings

We trained our network using MS-COCO dataset [6] for content images and WikiArt dataset [7] for style images, both of which contain approximately 80,000 images. We use Adam optimizer [8] with learning rate = 10^{-4} , lr-decay = $5 * 10^{-5}$ and a batch size of 8 content-style image pairs. Style loss weight λ_s is set to 10 after performing grid search on the hyperparameter space. For training, we first resized the images to 512 x 512 pixels followed by randomly cropping 256 x 256 pixels. As both the SAFIN module and the base network are fully convolutional, any input size can be used during the time of testing.

3.2. Effect of stylizing only LL component through SAFIN

[2] showed that stylizing LL component of the encoder affects the surface or texture information whereas high frequency components (LH, HL, HH) affect the edges. Therefore, we apply SAFIN module to both the low frequency and the high frequency components for an overall rich stylization. Figure 6 shows the effect of removal of SAFIN module from the high frequency components. It can be seen that stylizing high frequency components leads to better stylization because of sharper edges and boundaries.

More stylization results for comparison have been included (Fig. 7).

4. REFERENCES

- [1] Dae Young Park and Kwang Hee Lee, “Arbitrary style transfer with style-attentional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5880–5888.
- [2] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha, “Photorealistic style transfer



Fig. 6: Effect of SAFIN on high frequency components. The use of SAFIN on high frequency components leads to enhanced edge patterns thus making the output image sharper.

via wavelet transforms,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9036–9045.

- [3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [4] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena, “Self-attention generative adversarial networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 7354–7363.
- [5] Xun Huang and Serge Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.
- [6] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [7] Fred Phillips and Brandy Mackintosh, “Wiki art gallery, inc.: A case for critical thinking,” *Issues in Accounting Education*, vol. 26, no. 3, pp. 593–608, 2011.
- [8] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

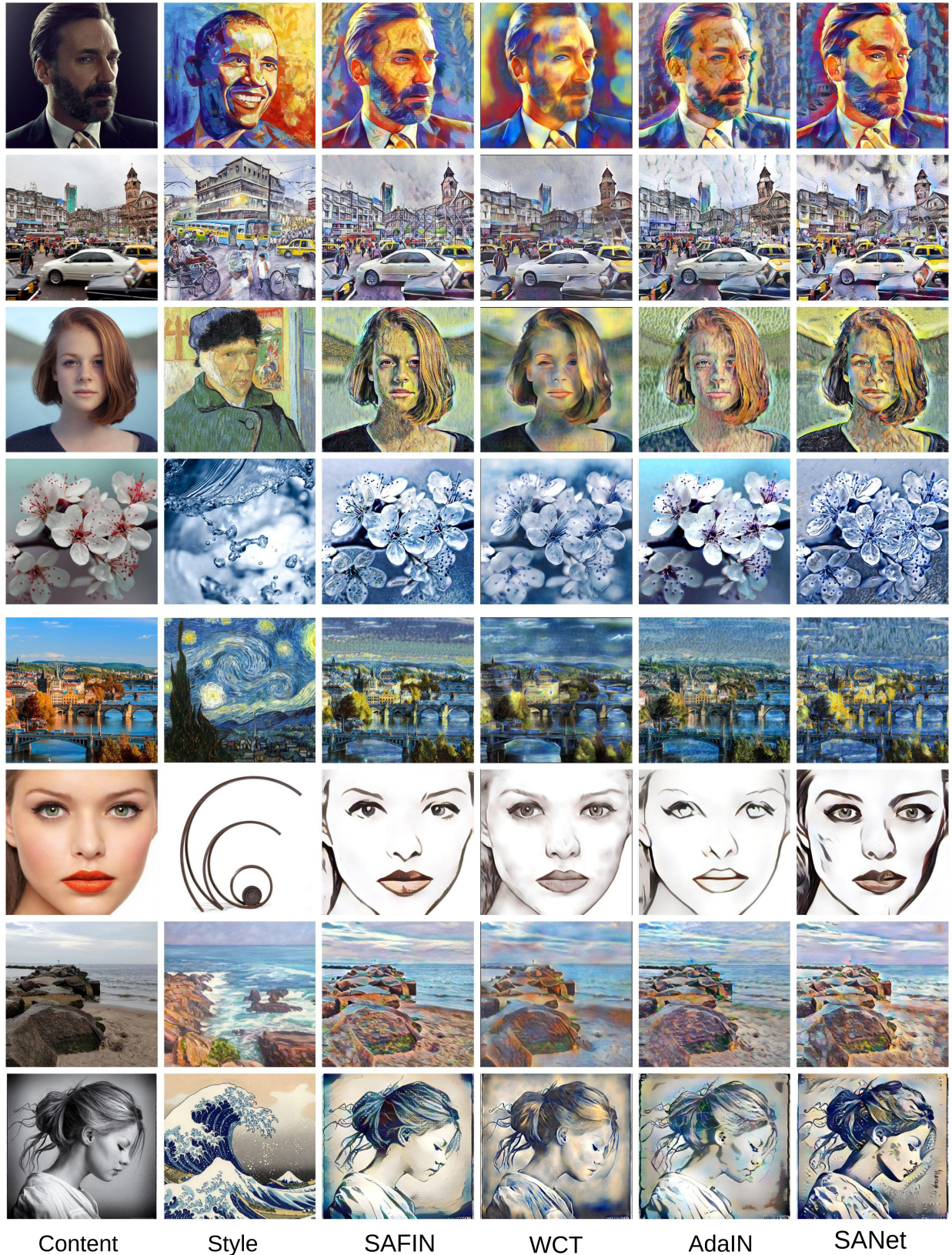


Fig. 7: Stylization results. Further stylization results for comparison (zoom in for greater clarity)