Dynamic Instance Normalization for Arbitrary Style Transfer

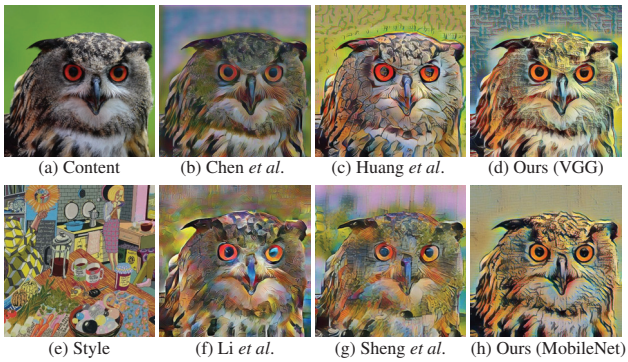
Yongcheng Jing,[1] Xiao Liu,[2] Yukang Ding,[2] Xinchao Wang,[3] Errui Ding,[2]
Mingli Song,[1]* Shilei Wen[2]
[1]Zhejiang University, [2]Department of Computer Vision Technology (VIS), Baidu Inc., [3]Stevens Institute of Technology
{ycjing, brooksong}@zju.edu.cn, {liuxiao12, dingyukang, dingerrui, wenshilei}@baidu.com, xinchao.wang@stevens.edu

Abstract

Prior normalization methods rely on affine transformations to produce arbitrary image style transfers, of which the parameters are computed in a pre-defined way. Such manually-defined nature eventually results in the high-cost and shared encoders for both style and content encoding, making style transfer systems cumbersome to be deployed in resource-constrained environments like on the mobile-terminal side. In this paper, we propose a new and generalized normalization module, termed as *Dynamic Instance Normalization (DIN)*, that allows for flexible and more efficient arbitrary style transfers. Comprising an instance normalization and a dynamic convolution, DIN encodes a style image into learnable convolution parameters, upon which the content image is stylized. Unlike conventional methods that use shared complex encoders to encode content and style, the proposed DIN introduces a sophisticated style encoder, yet comes with a compact and lightweight content encoder for fast inference. Experimental results demonstrate that the proposed approach yields very encouraging results on challenging style patterns and, to our best knowledge, for the first time enables an arbitrary style transfer using MobileNet-based lightweight architecture, leading to a reduction factor of more than twenty in computational cost as compared to existing approaches. Furthermore, the proposed DIN provides flexible support for state-of-the-art convolutional operations, and thus triggers novel functionalities, such as uniform-stroke placement for non-natural images and automatic spatial-stroke control.

Introduction

Image stylization has been a long-standing research topic. It has been studied in the domain of computer graphics, or more specifically, the area of *Non-Photorealistic Rendering (NPR)* (Gooch and Gooch 2001; Rosin and Collomosse 2012). In the field of computer vision, image stylization is studied as a generalized problem of texture synthesis (Efros and Leung 1999). Built upon the recent progress in visual texture modelling (Gatys, Ecker, and Bethge 2015) and image reconstruction (Mahendran and Vedaldi 2015), Gatys *et al*. (Gatys, Ecker, and Bethge 2016) propose to exploit *Convolutional Neural Networks (CNNs)* to render a content image in different styles, pioneering a new field called *Neural Style Transfer (NST)* (Jing et al. 2019).



(a) Content (b) Chen *et al.* (c) Huang *et al.* (d) Ours (VGG)
(e) Style (f) Li *et al.* (g) Sheng *et al.* (h) Ours (MobileNet)

Figure 1: Existing ASPM methods either barely transfer style to the target (*Chen et al., Huang et al.*), or produce distorted style patterns (*Li et al., Sheng et al.*) while relying on high-cost encoders. By contrast, the proposed DIN achieves superior performance using the same architecture (*Ours (VGG)*), and for the first time endows a much smaller lightweight network to transfer arbitrary styles (*Ours (MobileNet)*).

The inspiring work of Gatys *et al*. is, however, built upon an iterative image optimization in the pixel space, which turns out to be computationally expensive due to the online optimization. To address this efficiency issue, model-optimization-based NST algorithms are proposed, which optimize feed-forward models in an offline training manner. The earliest model-optimization-based NST algorithms, namely *Per-Style-Per-Model (PSPM)*, train separate style-specific models for each particular style, and are therefore burdensome to be adopted for real-world applications. (Johnson, Alahi, and Fei-Fei 2016; Ulyanov et al. 2016; Li and Wand 2016). To address this issue, *Multiple-Style-Per-Model (MSPM)* algorithms are proposed by incorporating multiple styles into one single model (Zhang and Dana 2017; Chen et al. 2017; Li et al. 2017b; Dumoulin, Shlens, and Kudlur 2017). Unfortunately, MSPM also suffers from

*Corresponding author

the inflexible binding between specific styles and a single model.

More recently, *Arbitrary-Style-Per-Model (ASPM)* algorithms are proposed to solve the aforementioned dilemma by exploiting one single model to transfer arbitrary new styles (Chen and Schmidt 2016; Huang and Belongie 2017; Sheng et al. 2018; Li et al. 2017c). Despite the great progress, ASPM algorithms still have limitations in handling complex style patterns and producing fine strokes. Moreover, they rely on high-cost and shared encoders for both style and content encoding, thus making the network cumbersome to be deployed on the mobile-terminal side. In this paper, we propose a novel module, termed as *Dynamic Instance Normalization (DIN)*, that allows for flexible and more efficient arbitrary style transfers. Unlike existing approaches that require a pre-defined way to compute parameters for their affine transformations for arbitrary style transfer, the proposed DIN introduces a generalized dynamic convolutional transformation, of which the parameters are learned adaptively for arbitrary stylization. In this way, DIN makes it possible to exploit a sophisticated style encoder to express complex and rich style patterns, and meanwhile preserves a compact and lightweight content encoder for fast inference. With the proposed DIN layer, we are able to conduct arbitrary style transfer, to our best knowledge, for the first time on a compact MobileNet-based architecture, resulting in plausible results with much lighter computational costs as compared to the state-of-the-art approaches. Furthermore, DIN supports various convolutional operations, and therefore enables novel transfer functionalities including automatic spatial-stroke control and uniform-stroke placement for non-natural images.

The proposed approach delivers gratifying visual results, especially for finer strokes and sharper details. An example that compares the visualizations of our approach and those of other ASPM methods is shown in Fig. 1, where the goal is to transfer the artistic style of a painting into a photo of an owl. In the stylized results of Chen *et al*. and Huang *et al*. (Fig. 1(b) and (c)), the target style (Fig. 1(e)) is not well reflected, since few style patterns are transferred. The results of Li *et al*. and Sheng *et al*. (Fig. 1(f) and (g)), on the other hand, are prone to distorted patterns and lack sharp details and fine strokes. By contrast, the proposed method is able to parse the challenging style patterns, like the wall brick patterns in Fig. 1(e), using the same architecture (Fig. 1(d)), and further for the first time enables arbitrary style transfers using lightweight MobileNet-based architecture (Fig. 1(h)). In sum, our contribution is an innovative dynamic instance normalization (DIN) layer that allows for more efficient and flexible arbitrary style transfers with favorable visual quality in generating challenging style patterns. This is achieved via a dynamic convolutional transformation with an elaborate style encoder and a lightweight content encoder. Experimental results demonstrate that the proposed method yields results superior to the state of the art both quantitatively and qualitatively, and meanwhile leads to a significant reduction of computation cost, at a factor of twenty.

Related Work

The goal of ASPM style transfer algorithms is to exploit one single trained model to migrate arbitrary artistic styles to a given photo with only one forward pass. There are two categories of ASPM algorithms in the literature, namely *Non-Parametric ASPM with Markov Random Fields* and *Parametric ASPM with Summary Statistics*.

The idea of non-parametric ASPM is to transfer artistic styles based on local patches. The first non-parametric ASPM, proposed in (Chen and Schmidt 2016), divides the content and style activations in the VGG feature space into a set of activation patches, and the target features are then obtained by mapping and swapping each content activation patch with the most similar style patch. By feeding the target features into the trained decoder, the stylized result can be produced. Another non-parametric ASPM in (Gu et al. 2018) further adds a constraint upon the algorithm of Chen *et al*., *i.e*., each patch is required to be mapped only once as possible as it can. In this way, their algorithm preserves better global style appearance as compared to (Chen and Schmidt 2016). Non-parametric ASPM enjoys favorable visual quality but suffers from the heavy computation burden brought by the mapping and swapping procedure.

Parametric ASPM improves the efficiency of non-parametric methods via global summary statistics matching. Specifically, (Huang and Belongie 2017) designs a novel *Adaptive Instance Normalization (AdaIN)* layer to explicitly transfer the channel-wise mean and variance statistics between style and content feature activations. Following their work, (Sheng et al. 2018) further extends AdaIN to multi-scale stylization for better visual quality.

Another line of parametric ASPM is based on *Whitening Instance Normalization (WIN)* proposed by (Li et al. 2017c). They find that whitening normalization in the VGG feature space can remove the style-related information and meanwhile preserve content structures of input images. Therefore, they first use whitening instance normalization to filter the style out of the content image, and then apply the coloring transforms to transfer the desired style patterns. Their proposed WIN-based approach successfully transfers arbitrary styles in a learning free manner. However, their whitening and coloring transforms is realized by matrix computations, which are computationally expensive. To address this issue, (Li et al. 2019) proposes to learn feed-forward networks to replace the matrix computations in (Li et al. 2017c).

These state-of-the-art ASPM algorithms still suffer from one major flaw: they all require high-cost VGG encoders. The one exception is the algorithm in (Shen, Yan, and Zeng 2018), which generates a whole 14-layer style-specific stylization network for every style, but leading to expensive cost of extra memory. Unlike these existing methods, our approach gets rid of the high-cost encoders and expensive memory, yet with superior quality in challenging styles.

Proposed Method

Revisiting Normalization Methods in NST

The development of NST is very related to the emergence of several novel normalization methods. Here, we revisit and

analyze existing normalization methods in the field of NST, which motivate the inspiration of the proposed approach.

Normalization for PSPM. The first emerged normalization method in NST, namely *instance normalization (IN)* or *contrast normalization* (Ulyanov, Vedaldi, and Lempitsky 2016), can be defined as follows:

$$\mathrm{IN}(\mathcal{F}_c) = \frac{\mathcal{F}_c - \mu(\mathcal{F}_c)}{\sigma(\mathcal{F}_c)}, \tag{1}$$

where $\mathcal{F}_c$ denotes the feature activation given the content image I_c as input. IN is first and primarily adopted in PSPM algorithms. It has been demonstrated that compared with *batch normalization*, IN is capable of better visual performance (Ulyanov, Vedaldi, and Lempitsky 2016; 2017) and can also achieve faster and better convergence (Huang and Belongie 2017).

Normalization for MSPM. Built upon the idea of IN, Dumoulin *et al*. further propose *conditional instance normalization (CIN)* (Dumoulin, Shlens, and Kudlur 2017), which is to scale and shift the activations in IN layer:

$$\mathrm{CIN}(\mathcal{F}_c, \mathcal{F}_s) = \gamma_{\mathcal{F}_s} \times \mathrm{IN}(\mathcal{F}_c) + \beta_{\mathcal{F}_s}, \tag{2}$$

where $\mathcal{F}_s$ represents the feature activations of the style image I_s, and $\gamma_{\mathcal{F}_s}$ and $\beta_{\mathcal{F}_s}$ are affine parameters corresponding to different styles. Dumoulin *et al*. find that by only changing learned γ and β in CIN layers with different style images, a single network can transfer multiple styles. Their proposed CIN is the earliest MSPM algorithm, and meanwhile achieves state-of-the-art performance even now.

Normalization for ASPM. Inspired by the success of CIN, Huang *et al*. propose to use adaptive affine parameters for arbitrary style transfer, through a novel *adaptive instance normalization (AdaIN)* layer (Huang and Belongie 2017). However, their "adaptive" affine parameters are computed in a manually defined manner, *i.e*., simply using the channel-wise mean μ and variance σ of style features, the most basic statistics, as the affine parameters in AdaIN layer. Their proposed AdaIN can be formulated as follows:

$$\mathrm{AdaIN}(\mathcal{F}_c, \mathcal{F}_s) = \sigma(\mathcal{F}_s) \times \mathrm{IN}(\mathcal{F}_c) + \mu(\mathcal{F}_s). \tag{3}$$

The intuition behind AdaIN is that the mean of VGG feature activations could encode different types of brushstrokes in a certain style, while the variance of the same feature activations could encode the amount of subtle style information, as explained in (Huang and Belongie 2017). Their algorithm achieves a reasonable performance in arbitrary image stylization. However, we believe that it is suboptimal to compute these "adaptive" affine parameters in such a manually defined way, thus leaving room for improvement.

More importantly, we find two implicit requirements for the favorable performance of AdaIN in arbitrary stylization:
1) The content and style encoders for producing $\mathcal{F}_c$ and $\mathcal{F}_s$ in Eq. 3 should be kept the same;
2) The network architecture of the encoders should be complex enough to extract high-level encoded features, like the VGG network.

Figure 2: Stylization results of using shared VGG encoders (2nd column), different VGG encoders (3rd column) and shared but much smaller encoders (4th column) for producing separate content and style features in AdaIN layer. *Enc* represents encoders.

To validate our first point, we show the comparison results of using shared VGG encoders (Fig. 2, 2nd column) and using different VGG encoders (Fig. 2, 3rd column) for producing $\mathcal{F}_c$ and $\mathcal{F}_s$. The results of using different VGG encoders are generated by setting the original style encoder in (Huang and Belongie 2017) as trainable while the content VGG encoder remains unchanged. Other experimental settings remain the same. The results with different encoders (Fig. 2, 3rd column), as can be observed, are inferior in visual quality with unexpected patterns, primarily because $\mathcal{F}_c$ and $\mathcal{F}_s$ are no longer in the same feature space.

Also, in the 4th column of Fig. 2, we try to use a smaller network architecture (Johnson, Alahi, and Fei-Fei 2016) to replace the original VGG encoders of AdaIN. In particular, the content and style encoders are shared, like the 2nd column in Fig. 2. The visual results of smaller encoders, as can be observed, are less appealing with many artifacts, which is consistent with our second point. We believe that the reason is: the way of manually defining the affine parameters as feature mean and variance in AdaIN needs a more meaningful high-level feature representation, which requires the encoder to be deep enough. This point is also partly observed in (Shen, Yan, and Zeng 2018).

These two implicit requirements reveal another major flaw of AdaIN: neither of the content and style encoder can be reduced to a lightweight one, which makes the network cumbersome for deployment in resource-constrained environments. *Whitening Instance Normalization (WIN)*, another group of ASPM, also has this issue, since whitening and coloring transforms also need meaningful features, as explained in related work.

Dynamic Instance Normalization

To address the aforementioned limitations in existing ASPM algorithms, we propose a new and generalized *Dynamic Instance Normalization (DIN)* layer for arbitrary image stylization. Instead of manually defining the way to compute the affine parameters so as to align the mean and variance, the simplest statistics, between content and style features for

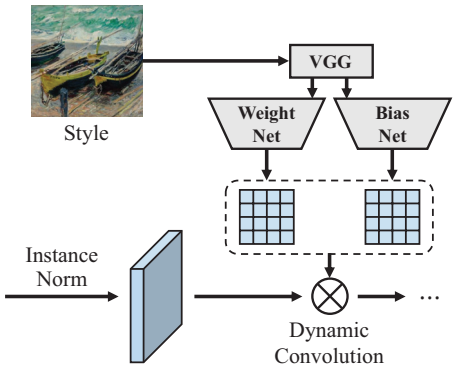


Figure 3: DIN layer consists of an instance normalization and a dynamic convolutional operation. Here, the convolution types include but are not limited to standard convolution, deformable convolution, and groupwise convolution.

arbitrary style transfer, we introduce a more generalized dynamic convolutional transformation, of which the parameters are adaptively changed in a learnable manner according to different styles, leading to a more accurate alignment of the real complex statistics of style features.

Given a pair of content image I_c and style image I_s as input, the proposed DIN layer can be modelled as:

$$\text{DIN}(\mathcal{F}_c, \mathcal{F}_s) = f[\mathcal{F}_s, \text{IN}(\mathcal{F}_c)], \tag{4}$$

where $\mathcal{F}_c$ and $\mathcal{F}_s$ are the corresponding feature representations of I_c and I_s, and f is a dynamic convolutional operation (Jia et al. 2016). Unlike standard convolutions of which the weight and bias are model parameters, in our dynamic convolution f, the weight and bias are dynamically generated by encoding different input style images.

Fig. 3 illustrates the basic structure of the proposed DIN, comprising an instance normalization, like previous CIN (Eq. 2) and AdaIN (Eq. 3), and a dynamic convolution, of which the parameters are adaptively changed by forwarding different input styles to separate weight and bias networks. In this way, the proposed DIN can be regarded to encode a given style image into learnable convolutional parameters via a sophisticated style encoder, and the content image is then stylized by the dynamic convolutional transformation. The weight and bias networks consist of two convolutional and adaptive pooling operations for handling arbitrary input sizes, of which the output size is set according to the type and hyperparameters of the dynamic convolution.

Compared with prior normalization methods, first, the proposed DIN model in Eq. 4 is much more generalized, including all the aforementioned existing normalization methods as special cases. Specifically, IN in Eq. 1 can be treated as a special case of our model in Eq. 4 where the weight and bias of f are set to 0 and 1, respectively. CIN in Eq. 1 corresponds to the case where both of the weight and bias in f are scalars. AdaIN in Eq. 3 is also a special case of the proposed DIN when the channel-wise variance and mean of style features are manually set as the weight and bias of f. The proposed DIN therefore can be treated as a generalized framework to normalization, under which existing normalization methods are taken as specific realizations, allowing for a larger search space for better optimized solutions and convergence.

Second, and more importantly, the proposed DIN does not have the aforementioned requirement of complex and shared encoders in previous ASPM methods, since we do not need to align the mean and variance of meaningful feature activations between content and style for stylization, thanks to our dynamic convolutional transformation. By contrast, the proposed approach enables a more sophisticated style encoder with dynamic convolution, so as to encode rich and complex style patterns adequately, yet using a more lightweight content encoder for faster inference, since the style-specific parameters for known styles can be stored in advance. As a result, the proposed DIN yields superior results in transferring challenging style patterns and fine strokes, and meanwhile conduct arbitrary stylizations with an over $20\times$ reduction in computation costs, as compared with the state of the art.

Third, the proposed DIN provides flexible support for a variety of convolutional operations, by simply changing the type of the dynamic convolution in Fig. 3. In particular, by incorporating some state-of-the-art convolutional operations, the proposed DIN is able to create novel functionalities in stylization. Here, we explain two variants of the proposed DIN as examples, which are deformable DIN and spatially-adaptive DIN.

1) Deformable Dynamic Instance Normalization. By using the deformable convolutional operation (Dai et al. 2017) for f, our algorithm achieves the first automatic spatial-stroke control in arbitrary style transfers. In deformable convolutions, the receptive fields and sampling locations can be adaptively adjusted according to the foreground objects' shape and scale. The deformable dynamic convolutional kernel, therefore, obtains the ability to stylize images in an attention-aware manner. For foreground and background objects, the deformable dynamic convolutional kernel would use different strokes according to the visual attention. As compared to existing approaches that randomly place different strokes across the whole image, the achieved automatic spatial-stroke control makes AI-created art much closer to human-created art.

2) Spatially-Adaptive Dynamic Instance Normalization. Prior ASPM algorithms based on IN suffer from another issue: They cannot generate proper strokes for uniform pixel areas of the input content image (Huang et al. 2018). The reason is that the convolutional output of the uniform pixel areas also has uniform values. After normalizing these uniform values through the widely adopted IN layer, the output activation would become all zeros, thereby preventing the proper stroke generations in the corresponding area. This limitation is especially serious for hand-crafted non-natural content images, where the uniform pixel areas are quite common. The issue can be addressed by setting the kernel size of f in the proposed DIN as the input feature map size, resulting in a special spatially-adaptive convolution similar to the operation used in (Park et al. 2019). In this way, the activation values for uniform pixel areas would not be uniform, leading to proper stroke generations in corresponding areas.

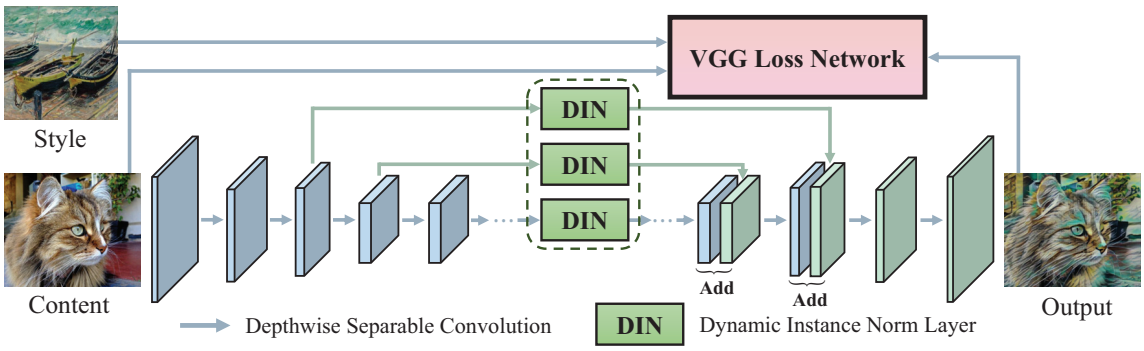


Figure 4: Network architecture of the proposed hierarchical lightweight arbitrary style transfer network based on MobileNetV1. DIN represents the proposed dynamic instance normalization layer as depicted in Fig. 3.

An Application of Dynamic Instance Normalization

We introduce here one major application of the proposed DIN in lightweight arbitrary style transfers for deployment in mobile and embedded devices. In particular, with the proposed DIN layer, we build the first MobileNet-based lightweight arbitrary stylization network in the literature, based on depthwise separable convolutions in MobileNetV1 (Howard et al. 2017).

Fig. 4 shows the proposed MobileNet-based network architecture, which mainly consists of three modules: image encoder, dynamic instance normalization layer and image decoder. The first module, image encoder, comprises one standard stride-1 convolutional layer, two stride-2 depthwise separable convolutional blocks and two stride-1 residual layers, inspired by the network design in (Johnson, Alahi, and Fei-Fei 2016). The image decoder is symmetric with the encoder, but with the pooling layers replaced by bilinear upsampling layers. In particular, the intermediate feature maps of the image decoder receive hierarchical normalized features from several separate DIN layers for additions.

The idea of this hierarchical DIN design comes from the traditional image processing algorithms, where images are generally decomposed hierarchically and each hierarchical level contains image information of a specific level. Since the features from a deep network also have a similar hierarchical structure, we follow and apply this idea in our network design. By hierarchically normalizing different levels of feature representations, the proposed DIN layers introduce multi-level style information for arbitrary style transfers. Although the single-level stylization with one single DIN layer also works in our experiment, the proposed hierarchical multi-level stylization can generate different scales of style elements better, and meanwhile preserve more content structures. More detailed architecture designs of the proposed network can be found in the supplementary material.

Experiments

Implementation Details

By default, the filter size of the proposed DIN layer is set to 1×1, considering the computational cost. We use the perceptual loss proposed in (Johnson, Alahi, and Fei-Fei 2016) as our content loss, and the BN-statistic loss proposed in (Li et al. 2017a) as our style loss, which are widely adopted in NST. Similar to prior works (Gatys, Ecker, and Bethge 2016; Huang and Belongie 2017; Jing et al. 2018), we use a pre-trained VGG-19 as our loss network. The content loss is computed at layer $\{relu4\_1\}$, while the style loss is computed at layer $\{relu1\_1, relu2\_1, relu3\_1, relu4\_1\}$ of the VGG network. During training, we adopt the Adam optimizer (Kingma and Ba 2015). The learning rates for both the image encoder and decoder are set to 0.0001. The weight and bias networks in DIN layers are set to have a $10\times$ learning rate for faster convergence. The training takes roughly one day on an NVIDIA Tesla V100 GPU.

Datasets

Our network is trained on $82,783$ content images from Microsoft COCO dataset (Lin et al. 2014), and $79,433$ style images from WikiArt (Nichol 2016). For inference, since there are no standard datasets in the current field of NST, we try to build here a new public testing dataset that contains diversified content and style images for evaluations. Specifically, we collect forty content images from *flickr.com*, containing roughly equivalent numbers of four categories: still life photos, portrait photos, animal photos, and landscape photos. For the style images, we select eight styles from *Google Arts & Culture*, including abstract, cubism, impressionism, surrealism, futurism, contemporary and expressionism. All the testing images are not used in training. The proposed testing dataset can be found in the supplementary material, which will also be publicly available.

Qualitative Evaluation

Fig. 5 demonstrates the qualitative results of the proposed DIN and other state-of-the-art ASPM algorithms (Li et al. 2017c; Huang and Belongie 2017; Sheng et al. 2018), which also use one single model for arbitrary style transfers. The comparison results are produced by using the official implementations with the default settings provided by the authors. All the testing content and style images are not used in training. In the last column of Fig. 5, we also produce the corresponding stylization results by using the popular PSPM algorithm of (Johnson, Alahi, and Fei-Fei 2016) as

Figure 5: Qualitative results of our proposed ASPM stylization algorithm and other methods. All the testing content and style images are not used in training. Addition experimental results can be found in the supplementary material.

Figure 6: Comparative results of the standard DIN and one of its variants, deformable DIN.

Table 1: Computation complexities (GFLOPs) of different network architectures with input size of 512×512.

Methods	Complexity	
	Encoder	**Decoder**
Johnson *et al.*	14.16	19.59
Li *et al.*	203.42	203.10
Huang *et al.*	63.44	63.36
Sheng *et al.*	63.44	63.36
Li and Liu *et al.*	63.44	63.36
Ours (MobileNet)	**3.62**	**3.72**

a gold standard, which trains separate style-specific models for different styles. In particular, to validate the effectiveness of the proposed DIN layer, we produce both the results of using the same VGG encoder as other ASPM algorithms (*Ours (VGG)*), and also the results of the proposed hierarchical MobileNet-based network (*Ours (MobileNet)*), which is about twenty times smaller than the VGG encoder.

The algorithm of Li *et al.*, as shown in Fig. 5, is not good at generating sharp details and fine strokes, due to its learning-free manner. Their results generally have distorted style patterns, *e.g.*, the background clutters in the 5$^{\text{th}}$ column of Fig. 5. By contrast, the algorithm of Huang *et al.* generates finer strokes; however, their algorithm is not effective at handling challenging style patterns, which is especially obvious in the 1$^{\text{st}}$ row, 6$^{\text{th}}$ column of Fig. 5, where very few style patterns are transferred. Also, their results have some similar repeated texture patterns among different styles, which might be caused by the suboptimal manually defined way for calculating parameters in AdaIN layer. The results of Sheng *et al.* also suffer from the issue of distorted patterns and lacking details. By contrast, with the same VGG encoder, our proposed DIN demonstrates superior performance in transferring challenging style patterns and meanwhile producing finer details (Fig. 5, 3$^{\text{rd}}$ column). Even if the encoder is reduced by a factor of twenty in complexity, the proposed DIN still achieves comparable quality (Fig. 5, 4$^{\text{th}}$ column) to the gold-standard PSPM algorithm of Johnson *et al.* (Fig. 5, 8$^{\text{th}}$ column).

Also, we show in Fig. 6 and Fig. 7 the results of the two aforementioned variants of the proposed DIN, respectively. Compared with standard DIN that uses standard con-

Figure 7: Comparative results of non-natural images produced by our standard DIN and the proposed spatially-adaptive DIN.

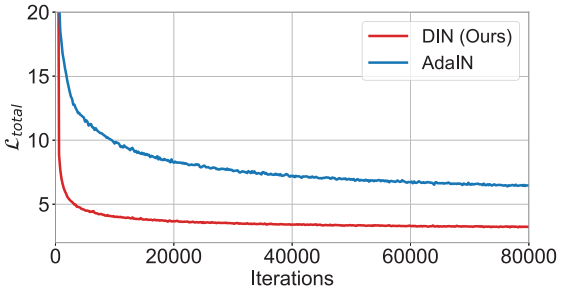


Figure 8: Comparing the training curves of the proposed DIN and AdaIN with the lightweight network architecture.

volutions, the proposed deformable DIN achieves automatic spatial-stroke control according to the visual attention, thus avoiding random stroke placement for fore- and background objects (*e.g.*, black and red strokes in the 2nd column of Fig. 6). The other variant, spatially-adaptive DIN, can generate proper strokes for uniform pixel areas in non-natural images, as shown in the last column of Fig. 7.

Quantitative Evaluation

Tab. 1 shows the comparison results among different architectures in terms of computation complexity, *i.e.*, the number of floating-point operations. Our proposed MobileNet-based network architecture outperforms other arbitrary stylization networks by a large margin in efficiency, even having much less computational cost than the PSPM network of (Johnson, Alahi, and Fei-Fei 2016). In particular, the additional weight and bias generators in our architecture bring little extra cost, *i.e.*, only 19.97MFLOPs given 512×512 input.

Also, to demonstrate the proposed DIN can lead to a better optimization, we show in Fig. 8 the quantitative comparison results in terms of training curves. Our proposed DIN, as can be noticed, achieves faster and better convergence under the same settings.

Ablation Study

To validate the proposed lightweight architecture in Fig. 4, we perform extensive ablation studies and demonstrate the

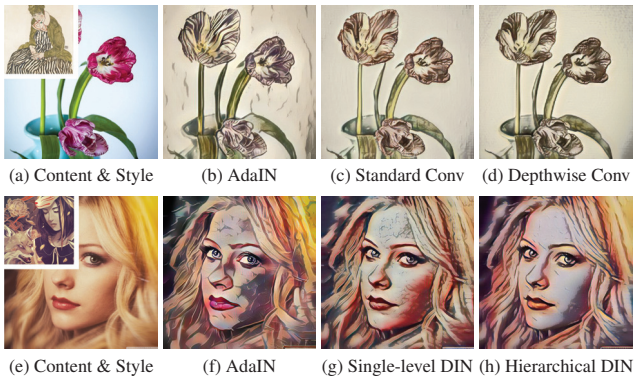


Figure 9: Results obtained using (c) standard convolutions, (d) depthwise convolutions, (g) single-level DIN, and (h) hierarchical DIN in the network architecture of Fig. 4, respectively.

corresponding results in Fig. 9. The first row of Fig. 9 shows the stylization results of using standard convolutions and depthwise separable convolutions in our network architecture. Despite the lower computational complexity, the visual quality using depthwise separable convolutions is still comparable to that using standard convolutions. In the second row of Fig. 9, we compare the results of using one single DIN layer and multiple hierarchical DIN layers depicted in Fig. 4. Our hierarchical design preserves finer structures of the input image, as can be observed in the human face of Fig. 9(h). Additional ablation studies including varying kernel sizes can be found in the supplementary material.

Discussions and Conclusions

In this paper, we introduce a novel dynamic instance normalization layer (DIN) for flexible and more efficient arbitrary style transfers. The proposed DIN allows for the use of an elaborate style encoder to encode rich style patterns and a lightweight content encoder for improved efficiency, thereby resolving the dilemma of redundant and shared encoders in previous methods. Experimental results demonstrate that the proposed approach achieves favorable performance against the state of the art, especially in transferring challenging style patterns while preserving a very light computational cost. In addition, by incorporating state-of-the-art convolutional operations, the proposed DIN is able to create novel effects in arbitrary style transfers, such as automatic spatial-stroke control.

In our future work, we would like to explore the use of DIN in other computer vision tasks beyond style transfer (Huang et al. 2018; Wang, Li, and Tao 2011; Dundar et al. 2018; Wang et al. 2014; Liu et al. 2019), as the proposed DIN can readily replace other normalization layers like CIN and AdaIN. In particular, we would like to introduce DIN to the context of domain adaption (Dundar et al. 2018), since style transfer is intrinsically a domain-adaption task (Li et al. 2017a; 2019).

Acknowledgement. This work is supported by National Key Research and Development Program (2016YFB1200203), National Natural Science Foundation of China (61976186), Key Research and Development Program of Zhejiang Province (2018C01004), and the Major Scientifc Research Project of Zhejiang Lab (No. 2019KD0AC01). Xinchao Wang is supported by the startup funding of Stevens Institute of Technology.

References

Chen, T. Q., and Schmidt, M. 2016. Fast patch-based style transfer of arbitrary style. In *NeurIPS Workshop on Constructive Machine Learning*.

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *CVPR*.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *ICCV*.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2017. A learned representation for artistic style. In *ICLR*.

Dundar, A.; Liu, M.-Y.; Wang, T.-C.; Zedlewski, J.; and Kautz, J. 2018. Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation. *arXiv preprint arXiv:1807.09384*.

Efros, A. A., and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *ICCV*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. Texture synthesis using convolutional neural networks. In *NeurIPS*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *CVPR*.

Gooch, B., and Gooch, A. 2001. *Non-photorealistic rendering*. Natick, MA, USA: A. K. Peters, Ltd.

Gu, S.; Chen, C.; Liao, J.; and Yuan, L. 2018. Arbitrary style transfer with deep feature reshuffle. In *CVPR*.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, X., and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*.

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal unsupervised image-to-image translation. In *ECCV*.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V. 2016. Dynamic filter networks. In *NeurIPS*.

Jing, Y.; Liu, Y.; Yang, Y.; Feng, Z.; Yu, Y.; Tao, D.; and Song, M. 2018. Stroke controllable fast style transfer with adaptive receptive fields. In *ECCV*.

Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Yu, Y.; and Song, M. 2019. Neural style transfer: A review. *TVCG*.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Li, C., and Wand, M. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*.

Li, Y.; Wang, N.; Liu, J.; and Hou, X. 2017a. Demystifying neural style transfer. In *IJCAI*.

Li, Y.; Chen, F.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017b. Diversified texture synthesis with feed-forward networks. In *CVPR*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017c. Universal style transfer via feature transforms. In *NeurIPS*.

Li, X.; Liu, S.; Kautz, J.; and Yang, M.-H. 2019. Learning linear transformations for fast arbitrary style transfer. In *CVPR*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Liu, M.-Y.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; and Kautz, J. 2019. Few-shot unsupervised image-to-image translation. In *ICCV*.

Mahendran, A., and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *CVPR*.

Nichol, K. 2016. Painter by numbers.

Park, T.; Liu, M.-Y.; Wang, T.-C.; and Zhu, J.-Y. 2019. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*.

Rosin, P., and Collomosse, J. 2012. *Image and video-based artistic stylisation*, volume 42. Springer Science & Business Media.

Shen, F.; Yan, S.; and Zeng, G. 2018. Neural style transfer via meta networks. In *CVPR*.

Sheng, L.; Shao, J.; Lin, Z.; Warfield, S.; and Wang, X. 2018. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *ArXiv e-prints*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*.

Wang, X.; Türetken, E.; Fleuret, F.; and Fua, P. 2014. Tracking interacting objects optimally using integer programming. In *ECCV*.

Wang, X.; Li, Z.; and Tao, D. 2011. Subspaces indexing model on grassmann manifold for image search. *TIP*.

Zhang, H., and Dana, K. 2017. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*.