A Content Transformation Block For Image Style Transfer

Dmytro Kotovenko   Artsiom Sanakoyeu   Pingchuan Ma   Sabine Lang   Björn Ommer
Heidelberg Collaboratory for Image Processing, IWR, Heidelberg University

Abstract

Style transfer has recently received a lot of attention, since it allows to study fundamental challenges in image understanding and synthesis. Recent work has significantly improved the representation of color and texture and computational speed and image resolution. The explicit transformation of image content has, however, been mostly neglected: while artistic style affects formal characteristics of an image, such as color, shape or texture, it also deforms, adds or removes content details. This paper explicitly focuses on a content-and style-aware stylization of a content image. Therefore, we introduce a content transformation module between the encoder and decoder. Moreover, we utilize similar content appearing in photographs and style samples to learn how style alters content details and we generalize this to other class details. Additionally, this work presents a novel normalization layer critical for high resolution image synthesis. The robustness and speed of our model enables a video stylization in real-time and high definition. We perform extensive qualitative and quantitative evaluations to demonstrate the validity of our approach.

1. Introduction

Style transfer renders the content of a real photograph in the style of an artist using either a single style sample [11] or a set of images [28]. Initial work on style transfer by Gatys et al. [9] proposed a method which exploits a deep CNN (Convolutional Neural Network) pretrained on a large dataset of natural images. Their costly computational optimization process has been replaced by an efficient encoder-decoder architecture in recent works [18, 28, 7, 2, 13] that efficiently generate the stylized output in a single feed-forward pass. While [18] has proven that an encoder-decoder architecture is both fast and effective for transferring style, it acts as a black-box model, lacking interpretability and accurate control of style injection: content transformation is performed indirectly, meaning there is no explicit control which part of the network carries out the stylization of photos and to what extend. To address this
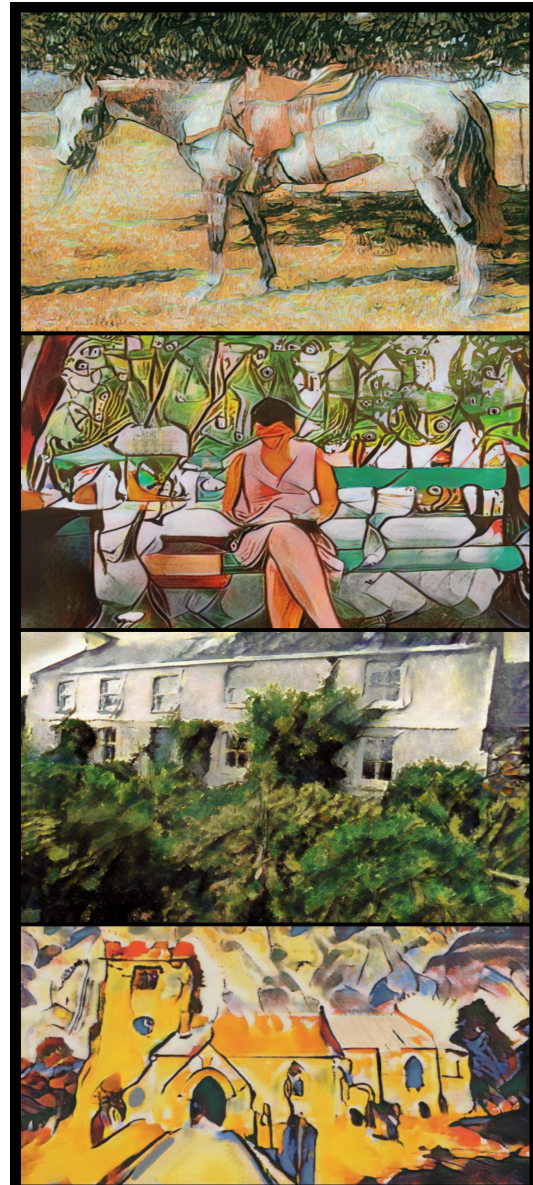

Figure 1. Examples of generated images using our approach in the style of (from top) Vincent van Gogh, Pablo Picasso, Paul Cezanne and Wassily Kandinsky. More stylization examples of images and videos can be found on the project page[2].

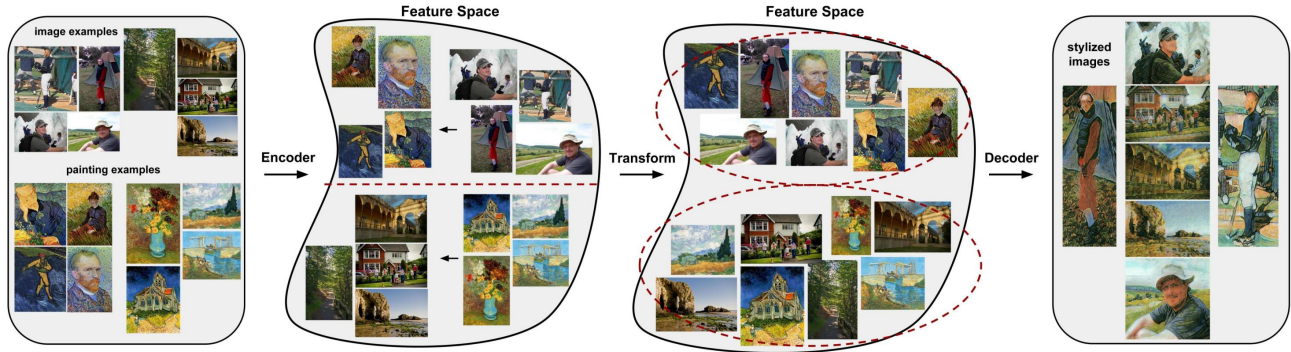[2] compvis.github.io/content-targeted-style-transfer/

Figure 2. Illustrates the role of the content transformation block. Real input images and painting examples displaying similar content are matched in the feature space. We utilize content similarity to learn how style explicitly alters content details.

issue, [28] introduced a fixpoint loss that ensures stylization has converged and reached a fixpoint after one feed-forward pass. This style-aware content loss forces the stylization to take place in the decoder. However, the main issue remains: The decoder alters style, synthesizes the stylized image, and upsamples it. All these individual tasks cannot be learned and controlled individually.

As a remedy, we introduce a novel *content transformation block* between encoder and decoder allowing control over stylization and achieving a style-aware editing of content images. We force the encoder to explicitly extract content information; the content transformation block $\mathcal{T}$ then modifies the content information in a manner appropriate to the artist's style. Eventually the decoder superimposes the style on the altered content representation. Our approach measures the content similarity between the content target image and stylized image before and after the transformation.

In contrast to previous work, stylization should be object specific and depending on the underlying object, the style transformation needs to adapt. The Cubist style of Picasso, for example, tends to reduce the human nose to a simple triangle or distorts the location of the eyes. Therefore, we further investigate, if we can achieve an object-specific alteration. We utilize similar content appearing in photographs and style samples to learn how style alters content details. We show that by using a prominent, complex, and diverse object class, i.e., persons, our model can learn how details are to be altered in a content-and style-aware manner. Moreover, the model learns to generalize beyond this one particular object class to diverse content. This is crucial to stylize also modern objects like computers which an artist like Monet never painted. In addition, we propose a local feature normalization layer to reduce the number of artifacts in stylized images, significantly improving results when moving to other image collections (i.e. from Places365 [34] to ImageNet [27]) and increasing the image resolution. To validate

the performance of our approach, we perform various qualitative and quantitative evaluations of stylized images and also demonstrate the applicability of our method to videos. Additional results can be found on the project page.

2. Related Work

Texture synthesis Neural networks were long used for texture synthesize [10]; feed-forward networks then enable a fast synthesis, however these methods often display a lack of diversity and quality [18, 30]. To circumvent this issue, [21] propose a deep generative feed-forward network, which allows to synthesize multiple textures within one single network. [12] has demonstrated how control over spatial location, color and across spatial scale leads to enhanced stylized images, where regions are altered by different styles; control over style transfer has been extended to stroke sizes [17]. [24] used a multiscale synthesis pipeline for spatial control and to improve texture quality and stability.

Separating content and style The integration of localized style losses improved the separation of content and style. In order to separate and recombine style and content in an image, works have utilized low-level features for texture transfer and high-level information to represent content using neural networks [11]. [6, 3, 8, 33] focused on distinguishing between different contents, styles and techniques in the latent space; to translate an image to another image is a vision problem, where the mapping between input and output image relies on aligned pairs. To avoid the need for paired examples, [35] presented an adversarial loss coupled with a cycle consistency loss to effectively assign two images. On the basis of [35], [28] has proposed an approach, where a style-aware content loss helps to focus on those content details relevant for a style. A combination of generative Markov random field (MRF) models and deep convolutional neural networks have been used for the task of

synthesizing content of photographs and artworks [20].

Real-time and super-resolution The processing time of style transfer and the resolution of images have been further addressed. Scholars aimed to achieve stylization in real time and in super-resolution using an unsupervised training approach, where either neural network features and statistics compute the acquired loss function [18] or a multi-scale network is employed [30]. To achieve a better quality for stylized images in high resolution, [32] propose a multimodal convolutional network, which performs a hierarchical stylization by utilizing multiple losses of increasing scales.

Stylizing videos While these works have approached the task of style transfer for input photographs, others concentrated on transferring artistic style to videos [25, 15, 28, 26], using feed-forward style transfer networks [4] or networks, which do not rely on optical flow at test time [15] to improve the consistency of stylization.

3. Approach

Let $\mathbf{Y}$ be a collection of images that defines a style. We extract the very essence of an artistic style presented in $\mathbf{Y}$ and learn to transfer it onto images from a different dataset $\mathbf{X}$, such as photos. This formulation resembles a typical unsupervised image translation problem, which requires a generator $\mathcal{G}$ (usually consisting of the encoder E and decoder D) and a discriminator $\mathcal{D}$ trained against each other: one mimics the target distribution $\mathbf{Y}$, the other one distinguishes between the authentic sample $y \in \mathbf{Y}$ and the stylized sample $D(E(x))$ for $x \in \mathbf{X}$. Hence, we can extract the style by solving the min-max optimization task for the standard adversarial loss:

$$\mathcal{L}_{adv} := \mathop{\mathbb{E}}_{y \sim \mathbf{Y}}[log(\mathcal{D}(y))]+ \\ \mathop{\mathbb{E}}_{x \sim \mathbf{X}}[log\left(1 - \mathcal{D}_s\left(D(E(x))\right)\right)] \quad (1)$$

Let s be additional content information that is easily available, i.e., we utilize a simple coarse scene label of the image x. Now the discriminator should not only discern real from synthesized art. It should also enforce that the scene information is retained in $D(E(x))$ by the stylization process,

$$\mathcal{L}_{cadv} := \mathop{\mathbb{E}}_{y \sim \mathbf{Y}}[log(\mathcal{D}(y))]+ \\ \mathop{\mathbb{E}}_{(x,s) \sim \mathbf{X}}[log\left(1 - \mathcal{D}\left(D(E(x))|s\right)\right)] \quad (2)$$

In contrast to a GAN framework that generates an image from a random vector z, style transfer not only requires to stylize a real input photograph x but also to retain the *content* of the input image after stylization. The simplest solution would be to enforce a per-pixel similarity between the input $x \sim \mathbf{X}$ and stylized image $\mathcal{G}(x)$:

$$\mathcal{L}_{pxl} := \mathop{\mathbb{E}}_{x \sim \mathbf{X}}[\|D(E(x)) - x\|_2^2]. \quad (3)$$

However, this loss alone would counter the task of stylization, since the image should not be the same afterwards on a per-pixel basis. Previous work [18, 9] has utilized a pretrained perceptual loss [29]. Since this loss is pretrained on an image dataset unrelated to any specific style, it cannot account for the characteristic way in which an artist alters content. Rather, we enforce the stylization to have reached a fixpoint, meaning that another round of stylization should not further alter the content. The resulting fixpoint loss measures the residual in the style-specific encoding space $E(\cdot)$,

$$\mathcal{L}_{FP} := \mathop{\mathbb{E}}_{x \sim \mathbf{X}}[\|E(D(E(x))) - E(x)\|_2^2]. \quad (4)$$

3.1. Content Transformation Block

While a painting of an artist is associated with one style, it is noticeable that style affects image regions differently: to emphasize the importance of an individual object, artists would use a more expressive brushstroke or deform it to a higher degree. Therefore, we do not only want to learn a simple stylization but a content-specific stylization. Thus each content detail must be stylized in a manner specific to this particular content category. This means that a stylized human figure should resemble how an artist has painted the figure in a specific style and not an arbitrary object, such as a vase or a chair. We enforce this capability by pulling images of similar content – but from different domains(art and photograph) – closer to each other in the latent space, while keeping dissimilar content images apart from each other. To be more specific, we force the content representation of an input photograph belonging to a specific class c to become more similar to the input painting's content representation of the same class. To achieve this, we introduce a content transformation block $\mathcal{T}$ transforming the output representation of the encoder E. We train this block in the adversarial fashion: the discriminator $\mathcal{D}_c$ has to distinguish between the representation of the real artworks' content and the transformed representation of the input photographs. But since we strive to obtain a content specific stylization, the discriminator $\mathcal{D}_c$ also has to classify the content class c of the artwork y and the content class of the input photograph x. Supplied with the content information c discriminator becomes more sensitive to content specific visual clues and enforces the content transformation block to mimic them in an artistic way.

$$\mathcal{L}_{adv-cont} := \mathop{\mathbb{E}}_{(y,c) \sim \mathbf{Y}}[log(\mathcal{D}_c(E(y)|c)))]+ \\ \mathop{\mathbb{E}}_{(x,c) \sim \mathbf{X}}[log\left(1 - \mathcal{D}_c\left(\mathcal{T}(E(x))|c\right)\right)] \quad (5)$$

In terms of neural architecture the $\mathcal{T}$ represents a concatenation of nine "residual blocks". Each block consists of six consecutive blocks with a skip connection: conv-
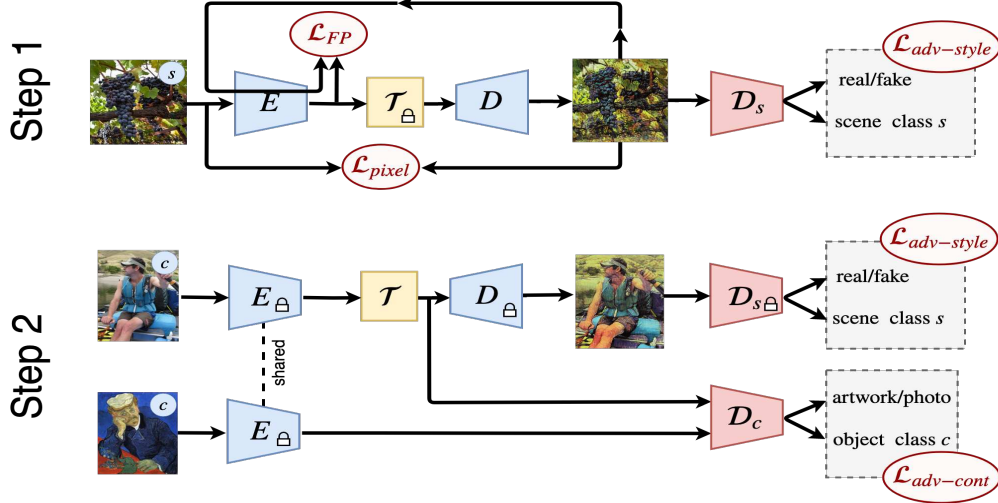
Figure 3. The two figures describe the two alternating training steps. The first step (top) is designated to obtain an artistic stylization while retaining the content information of the input photograph. The second step (bottom) trains the content transformation block $\mathcal{T}$ to alter the image content in a style-specific style. The lock sign indicates that the weights are fixed. See Approach section for further details.

layer, LFN-layer, `lrelu`-activation, conv-layer, LFN-layer, `lrelu`-activation.

3.2. Local Feature Normalization Layer

Many approaches using convolutional networks for image synthesis suffer from domain change (i.e. from photos of landscapes to faces) or synthesis resolution change. As a result, the inference size is often identical to the training size or the visual quality of the results deteriorates when switching to another domain. Reason being that instance normalization layers overfit to image statistics and the layer is not able to generalize to another image. We can improve the ability to generalize by enforcing stronger normalization through our local feature normalization layer. This layer normalizes the input tensor across a group of channels and also acts locally, not seeing the whole tensor but only the vicinity of the spatial location. Formally, for an input tensor $T \in \mathbb{R}^{B \times H \times W \times C}$, where B stands for the samples number, height H, width W and having C channels, we can define a *Local Feature Normalization Layer(LFN)* with parameters WS denoting spatial resolution of the normalization window and G - number of channels across which we normalize:

$$LFN(\cdot|WS, G) : \mathbb{R}^{B \times H \times W \times C} \longrightarrow \mathbb{R}^{B \times H \times W \times C}.$$

To simplify the notation, we first define a subset of the tensor T around (b, h, w, c) with a spatial window of size $WS \times WS$ and across a group of G neighbouring channels:

$$\mathbb{B}_{WS,G}(T, b, h, w, c) :=$$
$$\left\{ T(b, x, y, z) \middle\| \begin{array}{ccc} h - WS/2 & \leq x \leq & h + WS/2 \\ w - WS/2 & \leq y \leq & w + WS/2 \\ \lfloor \frac{c}{G} \rfloor G & \leq z \leq & \lfloor \frac{c}{G} \rfloor G + G \end{array} \right\}.$$
(6)

Finally, we can write out the expression for the Local Feature Normalization Layer applied to tensor T as:

$$LFN(T|WS, G)(b, h, w, c) :=$$
$$\gamma_c \frac{T(b, h, w, c) - \mathbf{mean}[\mathbb{B}_{WS,G}(T, b, h, w, c)]}{\mathbf{std}[\mathbb{B}_{WS,G}(T, b, h, w, c)]} + \beta_c. \quad (7)$$

In this equation, similar to the Instance Normalization Layer [31], parameters γ, $\beta \in \mathbb{R}^C$ denote vectors of trainable parameters and represent how to scale and shift each channel; those are learned jointly with other weights of the network via back-propagation. However, in practice the computation of **mean** and **std** of a large tensor could be a laborious task, so we compute these values only at the selected locations (b, h, w, c) and interpolate for others.

3.3. Training Details

The training dataset $\mathbf{X}$ is the union of the Places365 dataset [34] and the COCO dataset [23], such that for a tuple $(x, c, s) \in \mathbf{X}$ where x is a photograph, s is a scene class if x is from the Places dataset and c is a content class if x is from the COCO dataset. The second dataset $\mathbf{Y}$ contains tuples (y, c) where y is the artwork and c is the content

class. We focus on the content classes "person" and a negative class "non-person". The generator network consists of encoder E, transformer block $\mathcal{T}$ and decoder D. We utilize two conditional discriminators $\mathcal{D}_s$ and $\mathcal{D}_c$ - the former is applied to the input images and stylized outputs. The latter is applied to the content representation obtained by encoder E. Given this notation the losses become

$$
\begin{aligned}
\mathcal{L}_{adv-style} := &\underset{(y,c)\sim\mathbf{Y}}{\mathbb{E}}[log(\mathcal{D}_s(y))]+ \\
&\underset{(x,c,s)\sim\mathbf{X}}{\mathbb{E}}[log\left(1-\mathcal{D}_s\left(D(\mathcal{T}(E(x)))|s)\right)\right]
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
\mathcal{L}_{adv-cont} := &\underset{(y,c)\sim\mathbf{Y}}{\mathbb{E}}[log(\mathcal{D}_c(E(y)|c)))]+ \\
&\underset{(x,c,s)\sim\mathbf{X}}{\mathbb{E}}[log\left(1-\mathcal{D}_c\left(\mathcal{T}(E(x))|c)\right)\right]
\end{aligned}
\tag{9}
$$

$$
\mathcal{L}_{FP} := \underset{(x,c,s)\sim\mathbf{X}}{\mathbb{E}}[\|E(D(\mathcal{T}(E(x))))-E(x)\|_2^2].
\tag{10}
$$

Training procedure For variables $\theta_E, \theta_D, \theta_\mathcal{T}, \theta_{\mathcal{D}_c}, \theta_{\mathcal{D}_s}$ denoting parameters of the blocks $E, D, \mathcal{T}, \mathcal{D}_c, \mathcal{D}_s$. Training is performed in two alternating optimization steps.

The first step designated to obtain an accurate content extraction in encoder E and to learn a convincing style injection by decoder D.

$$
\begin{aligned}
\min_{\theta_E,\theta_D}\max_{\theta_{\mathcal{D}_s}}&\lambda_{\mathcal{L}_{pxl}}\mathcal{L}_{pxl}+ \\
&\lambda_{\mathcal{L}_{FP}}\mathcal{L}_{FP}+\lambda_{\mathcal{L}_{adv-style}}\mathcal{L}_{adv-style}
\end{aligned}
\tag{11}
$$

The second step is aimed to learn style-specific content editing by the block $\mathcal{T}$.

$$
\begin{aligned}
\min_{\theta_\mathcal{T}}\max_{\theta_{\mathcal{D}_c}}&\lambda_{\mathcal{L}_{adv-cont}}\mathcal{L}_{adv-cont}+ \\
&\lambda_{\mathcal{L}_{adv-style}}\mathcal{L}_{adv-style}
\end{aligned}
\tag{12}
$$

Please see Figure 3 illustrating the alternating steps of the training.

4. Experiments and Discussion

4.1. Stylization Assessment

To measure the quality of the generated stylizations we provide qualitative results of our approach and perform several quantitative experiments which we describe below.
Deception rate. This metric was introduced in [28] to asses how good the target style characteristics are preserved in the generated stylizations. A network pre-trained for artist classification should predict the artist which was used to generate the stylization. The deception rate is then calculated as the fraction of times the network predicted the correct artist. We report the deception rate for our and competing methods in Tab. 1 in the first column, where we can see that our approach outperforms other methods by a significant margin.

Figure 4. Can you guess which patches are real and which were generated by our approach? Each row contains three patches generated by our model and two real patches. Artists: (from the top) Cezanne, van Gogh, Claude Monet, Ernst Ludwig Kirchner and Berthe Morisot. The solution is provided on the last page.

Expert and non-expert score. We also perform human evaluation studies to highlight the quality of our stylization results. Given a content image patch, we stylize it with different methods and show results alongside a patch from a real painting to experts and non-experts. Both are asked to guess which one of the shown patches is real. The score is the fraction of times the stylization generated by this method was selected as the real patch. This experiment is performed with experts from art history and people without art education. Results are reported in Tab. 1.
Expert preference score. In addition, we asked art historians to choose which of the stylized images resemble the style of the target artist the most. Then the expert preference score (see Tab. 1) is calculated as the fraction of times the stylizations of the method was selected as the best among the others. The quantitative results in Tab. 1 show that both experts and non-experts prefer our stylizations in comparison to images obtained by other methods.
Content retention evaluation. To quantify how well the content of the original image is preserved, we stylize the ImageNet [27] validation dataset with different methods and compute the accuracy using pretrained VGG-16 [29] and ResNet-152 [14] networks averaged across 8 artists. Results presented in Tab. 2 show that the best classification score is achieved on stylizations by CycleGAN [36] and Gatys et al. [11], since both methods barely alter the content of the image. However, our main contribution is that we sig-

Method	Deception rate [28]	Non-Expert deception score	Expert deception score	Expert preference score
AdaIn [16]	0.067	0.033	0.016	0.019
WCT [22]	0.030	0.033	0.001	0.009
PatchBased[5]	0.061	0.118	0.011	0.038
Johnson et al. [18]	0.087	0.013	0.001	0.010
CycleGan [36]	0.140	0.026	0.031	0.010
Gatys et al. [11]	0.221	0.088	0.068	0.118
AST [28]	0.459	0.056	0.131	0.341
Ours	**0.582**	**0.178**	**0.220**	**0.456**
Wikiart test	0.6156	0.454	0.528	-
Photos	0.002	-	-	-

Table 1. A higher score indicates better stylization results. All scores are averaged over 8 different styles. The row "Wikiart test" [19] shows accuracy on real artworks from the test set. The deception rate for "Photos" shows how often photos were miss-classified by the network as real paintings of the target artist.

Network	Original photos	**Ours**	AST [28]	Gatys [11]	CycleGAN [36]
VGG-16	0.710	**0.016**	0.009	0.271	0.198
ResNet-152	0.783	**0.057**	0.032	0.389	0.341

Table 2. Top-1 classification accuracy on stylized images from validations set of ImageNet [27] using the networks pretrained on ImageNet. Note that the classification accuracy of our model is higher then the state-of-the art model [28]. We include results for [11] and [36], but they are not directly comparable to our method since they barely alter the content of the input image. In the second column we present the classification accuracy on the input photos.

Feature extractor	Photographs	**Ours**	Ours w/o $\mathcal{T}$	AST [28]
VGG-16	1.108	**0.756**	0.882	0.812
VGG-19	1.025	**0.724**	0.838	0.808

Table 3. The table summarizes RSSCD computed using different classification networks for different stylization methods. The score characterizes the content dissimilarity between real artworks and stylized images, relative to the average content dissimilarity between the artworks. The lower the better.

nificantly outperform the state-of-the-art AST [28] model on the content preservation task, while still providing more convincing stylization results, measured by the deception rate in Tab. 1.

Qualitative comparison. We compare our method qualitatively with existing approaches in Fig. 5. The reader may also try to guess between real and fake patches generated by our model in Fig.4. More qualitative comparisons between our approach and other methods are available in the supplementary material.

4.2. Ablation Study

4.2.1 Content Transformation

Relative style-specific content distance. To verify that the image content is transformed in a style-specific manner,
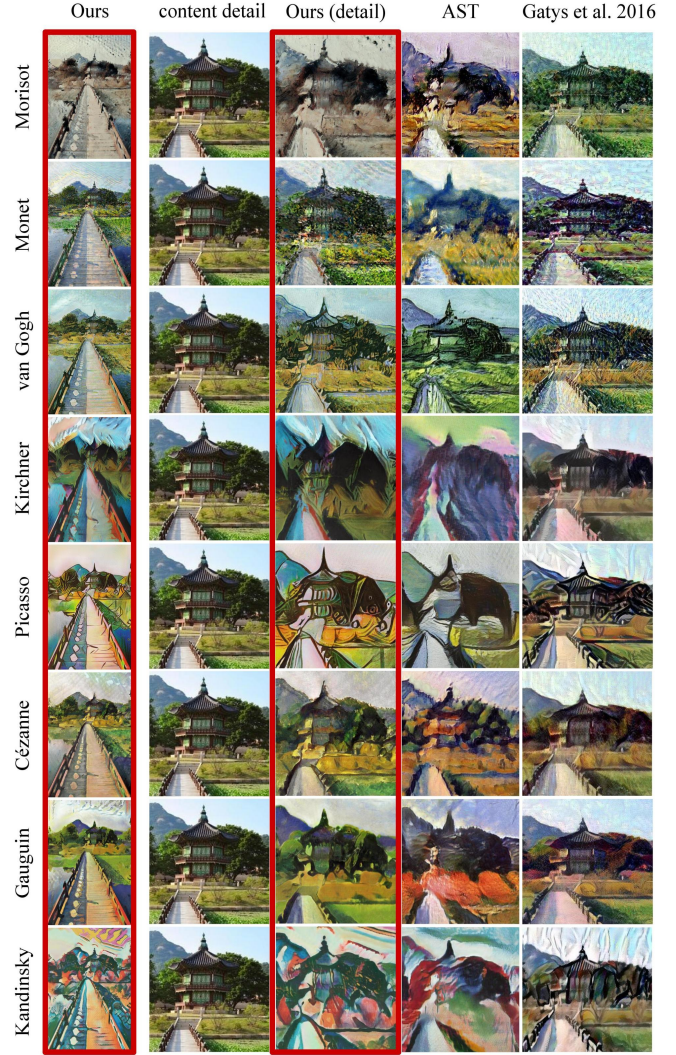

Figure 5. Qualitative comparison: the first column shows the entire image stylized by our approach (the real content image is provided on top), a detailed view is presented in the second. The third column shows the real photo detail respectively, while the last two display results obtained by other methods. Zoom in for a better view and details.

	Photographs	**Ours**	Ours w/o $\mathcal{T}$	AST [28]
Accuracy	0.953	**0.659**	0.598	0.548
Recall	0.978	**0.375**	0.209	0.109
Precision	0.927	0.864	**0.937**	0.893
F1-Score	0.954	**0.524**	0.343	0.195

Table 4. Results of the person detection on the stylized images from COCO dataset [23] using Mask-RCNN [1]. Columns from left to right: Person detection on photos; on stylized images by our method; on stylized images by our method without transformation block; on stylized images by AST [28].

we introduce a quantitative measure, called *relative style-specific content distance* (RSSCD). It measures the ratio

between the average distance of the generated image stylizations to the closest artworks and the average distance between all the artworks. Distances are computed using the features $\phi(\cdot)$ of the classification CNN pretrained on ImageNet. Then, RSSCD is defined as

$$\text{RSSCD} := \frac{\frac{1}{|Z_p|} \sum\limits_{z \in Z_p} \min\limits_{y \in Y_p} \|\phi(z) - \phi(y)\|_2}{\frac{1}{|Y_p||Y_n|} \sum\limits_{y_p \in Y_p, y_n \in Y_n} \|\phi(y_p) - \phi(y_n)\|_2},$$

Z_p denotes the set of stylizations of the positive content class (e.g., person), Y_p denotes the set artworks of the positive content class, and Y_n denotes all other artworks (see Fig. 6 for an illustration).

We report the RSSCD for our model with and without $\mathcal{T}$. For comparison we also evaluate the state-of-the-art approach AST [28]. Here, we use class "person" as the positive content class and two pretrained networks as content feature extractors $\phi(\cdot)$, namely VGG-16 and VGG-19 [29]. As can be seen in Tab. 3, the content transformation block significantly decreases the distance between the stylized images and original van Gogh paintings, proving its effectiveness.

We measure how well our model retains the information present in the selected "person" class and compare it to both the model not using $\mathcal{T}$ and to the AST [28]. We run the Mask-RCNN detector [1] on images from the COCO [23] dataset stylized by different methods and compute the accuracy, precision, recall and F1-score. From results ins Tab. 4 we conclude that the proposed block $\mathcal{T}$ helps to retain visual details relevant for the "person" class.

In Fig. 7 we show stylizations of our method with and without content transformation block. We recognize that applying the content transformation block alters the shape of the human figures in a manner appropriate to van Gogh's style resulting in curved forms (cf. the crop-outs from original paintings by van Gogh provided in the 4th column of Fig. 7). For small persons, the artist preferred to paint homogeneous regions with very little texture. This is apparent, for example, in the stylized patches in row one and six. Lastly, while van Gogh's self-portraits display detailed facial features, in small human figures he tended to remove them (see our stylizations in 3rd and 4th rows of Fig. 7). This might be due to his abstract style, which included a fast-applied and coarse brushstroke.

4.2.2 Generalization Ability

The transformer block $\mathcal{T}$ learns to transform content representation of the photographs of class "person" in such a way that it becomes indistinguishable of the content representation of artworks of the same class. Though transformation has been learned for one only class "person" it
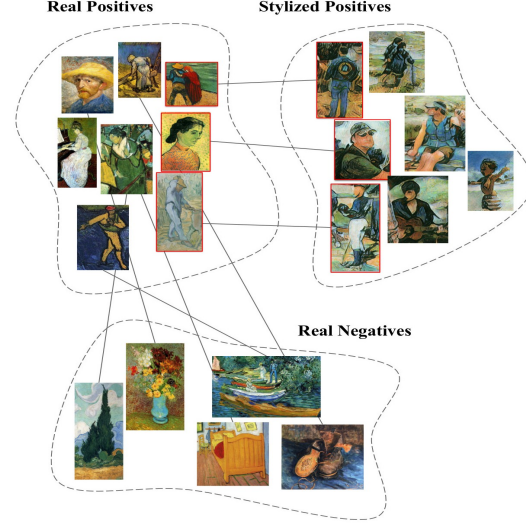

Figure 6. Illustration of the RSSCD measure. Real positive, real negative and stylized positive images are mapped into the feature space using deep CNN. We then compute the average distance from a stylized positive to the closest real positive and divide it by the average distance between real positives and negatives. Positive images correspond to the class "person".

Method	Deception rate [28] "person"	Deception rate non-"person"	Non-Expert deception score "person"	Non-Expert deception score non-"person"
AST	0.398	0.485	0.016	0.086
Ours w/o $\mathcal{T}$	0.521	0.541	0.127	0.143
Ours	**0.618**	**0.563**	**0.210**	**0.165**

Table 5. Stylization quality for different content classes. Our model has significantly improved stylization quality compared to the state-of-the-art AST[28] model. The $\mathcal{T}$ block improves the deception rate and preference score on both classes. The higher the better.

can still generalize to other classes. To measure this generalization ability we compute the deception rate[28] and non-expert deception scores on stylized patches for classes "person" and non-"person" separately. The evaluation results are provided in Tab. 5 and indicate improvement of the stylization quality for unseen content.

4.2.3 Artifacts Removal

To verify the effectiveness of the local feature normalization layer (LFN layer), we perform a visual inspection of learned models and notice prominent artifacts illustrated in Fig. 8. We can observe that especially for plain regions with little structure, the model without a LFN layer often produces unwanted artifacts. In comparison, results obtained with an LFN layer show no artifacts in the same regions. Notably, for a model without an LFN layer the number of artifacts increases proportionally to the resolution of the stylized image.

content   Ours w/o **T**   **Ours**   van Gogh (detail)
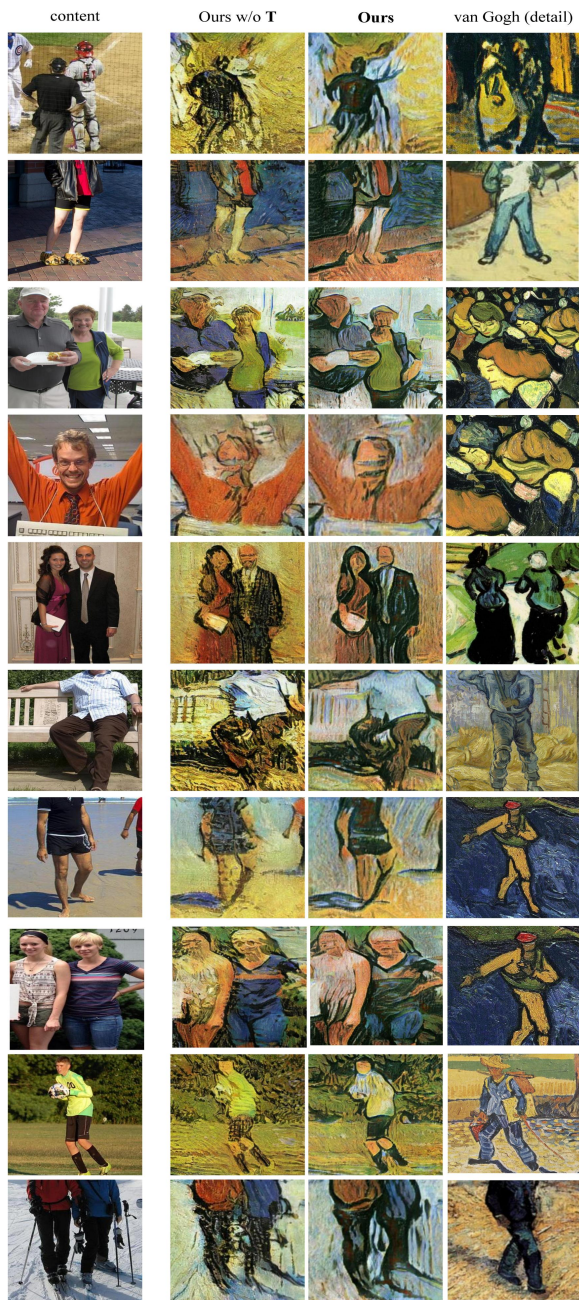
Figure 7. Shows the impact of the content transformation block: column two shows results without using a content transformation block. The third displays identical details with a content transformation block, emphasizing outlines of figures in a manner appropriate to van Gogh's style. The last column provides details from the artist's paintings to highlight the validity of our approach. Best seen on screen and zoomed in.

5. Conclusion

We introduced a novel content-transformation block designed as a dedicated part of the network to alter an object

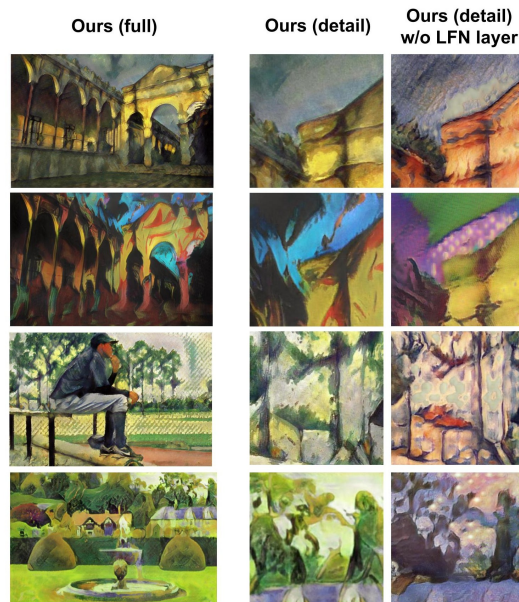**Ours (full)**   **Ours (detail)**   **Ours (detail) w/o LFN layer**

Figure 8. From left to right: stylized image with LFN layer, detail of the image, same region for the model without LFN layer. Styles from top to bottom: Cezanne, Kirchner, Cezanne, Paul Gauguin. Local feature normalization significantly reduces the number of artifacts by normalizing the network activation statistics.

in a content-and style-specific manner. We utilize objects from the same class in content and style target images to learn how content details need to be transformed. Experiments show that from only one complex object category, our model learns how to stylize details of content in general and thus improves the stylization quality for other objects as well. In addition, we proposed a local feature normalization layer, which significantly reduces the number of artifacts in stylized images, especially when increasing the image resolution or applying our model to previously unseen image types (photos of faces, road scenes etc.). The experimental evaluation showed that both art experts and persons without specific art education preferred our method to others. Our model outperforms existing state-of-the-art methods in terms of stylization quality in both objective and subjective evaluations, also enabling a real-time and high definition stylization of videos.

Acknowledgements

Solution to Figure 4:
Cezanne: fake, real, fake, fake, real
van Gogh: real, fake, real, fake, fake
Monet: fake, real, fake, real, fake
Kirchner: real, fake, fake, fake, real
Morisot: fake, real, fake, real, fake.

References

[1] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 6, 7

[2] M. Babaeizadeh and G. Ghiasi. Adjustable real-time style transfer. *CoRR*, abs/1811.08560, 2018. 1

[3] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *Advances in Neural Information Processing Systems*, pages 3846–3854, 2016. 2

[4] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stylebank: An explicit representation for neural image style transfer. In *Proc. CVPR*, volume 1, page 4, 2017. 3

[5] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016. 6

[6] J. Collomosse, T. Bui, M. J. Wilber, C. Fang, and H. Jin. Sketching with style: Visual search with sketches and aesthetic context. In *ICCV*, pages 2679–2687, 2017. 2

[7] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. *Proc. of ICLR*, 2017. 1

[8] P. Esser, E. Sutter, and B. Ommer. A variational u-net for conditional appearance and shape generation. *CoRR*, abs/1804.04694, 2018. 2

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 1, 3

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *arXiv preprint arXiv:1505.07376*, 12, 2015. 2

[11] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE, 2016. 1, 2, 5, 6

[12] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[13] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei. Characterizing and improving stability in neural style transfer. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4087–4096, 2017. 1

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 5

[15] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu. Real-time neural style transfer for videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7044–7052. IEEE, 2017. 3

[16] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 6

[17] Y. Jing, Y. Liu, Y. Yang, Z. Feng, Y. Yu, D. Tao, and M. Song. Stroke controllable fast style transfer with adaptive receptive fields. *arXiv preprint arXiv:1802.07101*, 9, 2018. 2

[18] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 1, 2, 3, 6

[19] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013. 6

[20] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016. 3

[21] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Diversified texture synthesis with feed-forward networks. In *Proc. CVPR*, 2017. 2

[22] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pages 385–395, 2017. 6

[23] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4, 6, 7

[24] E. Risser, P. Wilmot, and C. Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017. 2

[25] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, pages 26–36. Springer, 2016. 3

[26] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos and spherical images. *International Journal of Computer Vision*, pages 1–21, 2018. 3

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 2, 5, 6

[28] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer. A style-aware content loss for real-time hd style transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3, 5, 6, 7

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 5, 7

[30] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016. 2, 3

[31] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 4

[32] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 7, 2017. 3

[33] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. J. Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *ICCV*, pages 1211–1220, 2017. 2

[34] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2, 4

[35] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 2

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, 2017. 5, 6