DeSmoothGAN: Recovering Details of Smoothed Images via Spatial Feature-wise Transformation and Full Attention

Yifei Huang
East China Normal University
yifeihuang17@gmail.com

Chenhui Li*
East China Normal University
chli@cs.ecnu.edu.cn

Xiaohu Guo
The University of Texas at Dallas
xguo@utdallas.edu

Jing Liao
City University of Hong Kong
jingliao@cityu.edu.hk

Chenxu Zhang
The University of Texas at Dallas
chenxu.zhang@utdallas.edu

Changbo Wang*
East China Normal University
cbwang@cs.ecnu.edu.cn

ABSTRACT

Recently, generative adversarial networks (GAN) have been widely used to solve image-to-image translation problems such as edges to photos, labels to scenes, and colorizing grayscale images. However, how to recover details of smoothed images is still unexplored. Naively training a GAN like pix2pix causes insufficiently perfect results due to the fact that we ignore two main characteristics including spatial variability and spatial correlation as for this problem. In this work, we propose DeSmoothGAN to utilize both characteristics specifically. The spatial variability indicates that the details of different areas of smoothed images are distinct and they are supposed to be recovered differently. Therefore, we propose to perform spatial feature-wise transformation to recover individual areas differently. The spatial correlation represents that the details of different areas are related to each other. Thus, we propose to apply full attention to consider the relations between them. The proposed method generates satisfying results on several real-world datasets. We have conducted quantitative experiments including smooth consistency and image similarity to demonstrate the effectiveness of DeSmoothGAN. Furthermore, ablation studies are performed to illustrate the usefulness of our proposed feature-wise transformation and full attention.

CCS CONCEPTS

• **Computing methodologies → Image processing**; *Neural networks.*

KEYWORDS

detail recovering; spatial feature-wise transformation; full attention; generative adversarial network

ACM Reference Format:
Yifei Huang, Chenhui Li, Xiaohu Guo, Jing Liao, Chenxu Zhang, and Changbo Wang. 2020. DeSmoothGAN: Recovering Details of Smoothed Images via Spatial Feature-wise Transformation and Full Attention . In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413958

1 INTRODUCTION

The modern beauty smooth cameras generate attractive and mysterious selfies, due to the image modifications performed by the black box beauty algorithms. While such algorithms have helped us create pleasing beautifications, if used without the viewer's knowledge, they can cause serious problems such as producing fake impressions before dating or affecting photo forensics [9] when handling criminal cases. To avoid such problems, it is important for us to come up with a toolkit to reverse this process and one of its key procedures is to recover the details of your selfies that are lost due to the smoothing operations. In multimedia, it is related to how to recover details of all kinds of smoothed images including selfies as illustrated in Figure 1. The goal of this work is to explore the foundational techniques about how to recover details of smoothed images. Such detail recovering techniques can even help artists repair murals [28]. It is worth mentioning that recovering details of smoothed images is different from super resolution [6] since the missing details between smoothing algorithms and downsampling operations are different.

Recently, several image-to-image translation problems [16, 21, 23, 29, 40, 44, 53, 55, 56] have been explored with the popular GAN [13]. For example, pix2pix [23], as a general image-to-image translation framework, generates amazing results in edges to photos, labels to scenes, and so on. In contrast to pix2pix that uses a single phase, Zhang et al. [53] proposed to fill large holes in natural images progressively in multiple phases based on the observation that humans are good at thinking missing contents from the exterior to the interior of the hole. However, the boundary areas between the holes and the original image are often unsatisfactory. A fusion block is proposed in [16] in order to provide a smooth transition at the boundary area by generating a flexible alpha composition map. As for faces manipulated by Photoshop, Wang et al. [44] not only developed a network to classify whether the face images are manipulated or not, but also proposed a local warping field prediction network to recover the unmanipulated faces.

However, it remains unclear and challenging about how to recover details of smoothed images. Designing several functions manually to recover different objects is extremely complex owing to the boundless details. Instead, we propose to learn this kind of

[*] Chenhui Li and Changbo Wang are corresponding authors.

MM '20, October 12–16, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413958

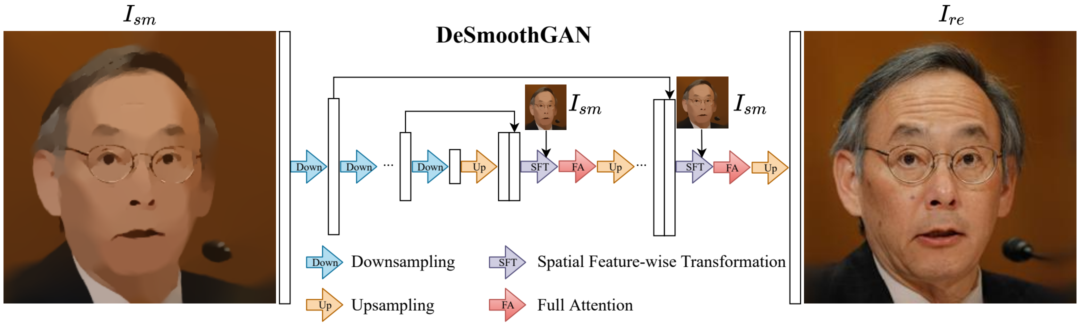


Figure 1: Illustration of recovering details of the smoothed images and our proposed DeSmoothGAN, where the spatial feature-wise transformation and full attention modules are placed in the upsampling phase.

function through the DeSmoothGAN, which is built on pix2pix [23] as a general image-to-image translation framework, and consists of a generator to recover details of the input smooth image and a discriminator to encourage the generator to recover as more details as possible. One key for employing a GAN-based solution lies in an appropriate design of the network architecture. Considering the specificity of smoothed images, we design the corresponding specific blocks to enhance performance. Especially, we conclude two main characteristics consisting of spatial variability and spatial correlation for smoothed images. The spatial variability drives us to recover the distinct areas of one smoothed image differently. Therefore, we propose to perform spatial feature-wise transformation to achieve this goal. The spatial correlation motivates us to consider the relations between areas when recovering a smoothed image. As a result, we propose to design full attention to utilize the relations between different areas. Our DeSmoothGAN follows the encoder-decoder structure [14] as shown in Figure 1. We place the spatial feature-wise transformation and the full attention modules in the upsampling phase because we expect the details are recovered from coarse to fine in the progressive upsampling phase.

In order to evaluate our DeSmoothGAN, we propose two quantitative measures. One is the smooth consistency (SC) which means if we smooth the generated image, the smoothed result should be as similar as the input image. The other is the image similarity metric measuring the similarity between the generated image and the ground truth. Our contributions can be summarized as follows:

- By exploiting the spatial variability that details of different areas of smoothed images are varied, we propose to utilize spatial feature-wise transformation to recover different details correspondingly.
- Considering the spatial correlation that details of different areas of smoothed images are related to each other, we propose to utilize full attention to take advantage of the relations.

2 RELATED WORK

In this section, we review the research works related to feature-wise transformations and attention mechanisms separately.

2.1 Feature-wise Transformation

Feature-wise transformation [7] indicates that we perform transformation at the feature level. Specifically, the affine transformation ($y = \gamma * x + \beta$) has been widely used considering effectiveness and efficiency, where x and y represent features and γ and β represent the learned parameters. The γ and β are used to scale and shift learned features, respectively. According to different tasks, the parameters γ and β are learned in different ways.

As for visual reasoning [36], a visual network that processes images and a linguistic network that processes text-based questions are often used. The γ and β that are used to transform the feature x of the visual network in [36] are learned from the linguistic network. Later, instead of generating γ and β all at once from the linguistic network, Strub et al. [39] employ an attention mechanism to generated γ and β in a multi-hop way. Furthermore, if we pretrain the visual network and train the linguistic network while freezing all parameters of the visual network except for γ and β, the visual reasoning ability of the visual network can be enhanced [5].

As for style transfer [11], γ and β are used to represent different style images and stored in an embedding table as in [8]. Later, Ghiasi et al. [12] proposed to predict γ and β using an auxiliary style prediction network that is trained jointly with a primary network that performs style transfer. What is more, Huang et al. [19] showed that the primary network itself can generate γ and β and performs style transfer simultaneously.

As for generative modeling, one form of conditioning in the autoregressive PixelCNN [41] is to only generate β while setting $\gamma = 1$. The StyleGAN [25] utilizes a mapping network including several fully connected layers to generate γ and β. A similar mappling network used in [20] also generates γ and β based on the style code in order to perform multimodal image-to-image translation.

The γ and β we discuss above are one-dimensional. However, when processing two-dimensional image data, an inherent limitation of one-dimensional γ and β is that it treats all pixels equally and uses the same weight for them. It does not work as well as we expect in some tasks below. In order to recover realistic textures from low-resolution images with different styles, Wang et al. [46] proposed to generate two-dimensional γ and β based on semantic guidance. As for generating photorealistic images from semantic layouts, Park et al. [35] showed that generating two-dimensional

γ and β is the key to generate satisfying images successfully. A bidirectional spatial feature-wise transformation (two-dimensional γ and β) is also used in [1] to achieve a guided image-to-image translation.

In this work, as for recovering details of smoothed images, different areas are supposed to be recovered differently. Therefore, we choose to perform spatial feature-wise transformation (two-dimensional γ and β) to achieve pleasing results.

2.2 Attention Mechanisms

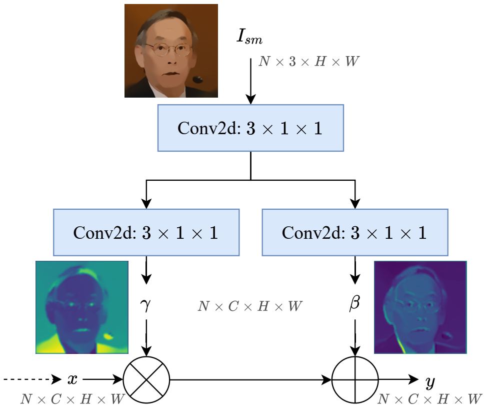
Attention mechanisms have been widely used in processing sequences [2, 43] due to the fact that it can model long-distance relations. Especially, Bahdanau et al. [2] are the first to utilize attention in a Recurrent Neural Network [15] to enhance alignment machine translation. Later, the performance of machine translation is improved by Transformer architecture [42] with self-attention.

With the success above in natural language processing, attention mechanisms are also applied in several visual tasks. For example, in order to generate detailed and satisfying images, Zhang et al. [52] proposed a self-attention generative adversarial network that models the long-range relation between different pixels. Hu et al. [18] reweighed the channel feature maps by aggregating information from the whole feature maps and calculating their relation in order to improve the performance of image classification and object detection. Furthermore, Hu et al. [17] enhanced this idea by introducing a general gather-excite operator. Instead of only refining channel feature maps, Woo et al. [48] also applied the spatial attention to refine feature maps after channel attention. However, how to combine channel attention and spatial attention matters. Rather than utilizing channel attention and spatial attention sequentially, Park et al. [34] proposed to perform channel attention and spatial attention in parallel. Similarly, Chen et al. [4] proposed double attention networks consisting of feature gathering and feature distribution, which results in better performance than non-local neural networks [45]. In non-local neural networks [45], several non-local residual blocks combining self-attention with convolutional modules are added to the main architecture. However, one limitation of [45] is that it needs ImageNet pretraining. Therefore, attention augmented convolution networks [3] concatenating features maps from both convolution and self-attention were proposed to solve this problem.

In this work, we target at recovering details of smoothed images. When recovering one area of a smoothed image, it is necessary to consider the relations between this and the other areas in the same image. Therefore, we propose full attention to achieve this goal.

3 METHOD

The goal of this work is to recover details of smoothed images. Considering the extreme complexity to design rules manually for different objects, we propose to learn this recovering process in a data-driven way. Nowadays, one typical solution to apply data-driven ideas is to train a GAN on the corresponding dataset. However, the results of a trained GAN are still heavily dependent on the network architecture and the loss functions. In this work, we specifically design spatial feature-wise transformation as shown in Section 3.1 and full attention as described in Section 3.2 to enhance performance. Then, we describe the loss functions in Section 3.3.



I_{sm}
$N \times 3 \times H \times W$
Conv2d: $3 \times 1 \times 1$
Conv2d: $3 \times 1 \times 1$
Conv2d: $3 \times 1 \times 1$
γ
$N \times C \times H \times W$
β
x
$N \times C \times H \times W$
y
$N \times C \times H \times W$

Figure 2: Spatial feature-wise transformation (SFT). The input feature x is scaled by γ and shifted by β spatially. The symbol $\otimes$ denotes element-wise matrix multiplication and the symbol $\oplus$ represents element-wise matrix addition. The output feature y will be further processed by full attention.

3.1 Spatial Feature-wise Transformation

Feature transformation has been a common and popular technique in several areas [19, 35, 36], and pix2pix [23] as a general image-to-image translation framework is no exception. The popular feature transformation can be represented as follow:

$$y = \gamma * x + \beta, \tag{1}$$

where $x \in \mathbb{R}^{N \times C \times H \times W}$ and $y \in \mathbb{R}^{N \times C \times H \times W}$ describe the input feature and the output feature, γ and β denote the transformation parameters.

There are two main limitations for pix2pix if we naively apply it to recover details of smoothed images. One limitation of pix2pix is that it treats all pixels equally. It means $\gamma \in \mathbb{R}^{N \times C \times 1 \times 1}$ and $\beta \in \mathbb{R}^{N \times C \times 1 \times 1}$. This is somewhat inappropriate for recovering details of smoothed images due to the fact that the details of different areas are different and thus, different areas of smoothed images are supposed to be recovered differently. Therefore, we propose to perform spatial feature-wise transformation (SFT) to compensate for spatial variability. SFT means $\gamma \in \mathbb{R}^{N \times C \times H \times W}$ and $\beta \in \mathbb{R}^{N \times C \times H \times W}$. The other limitation is that the transformation parameters γ and β are learned to be optimal over the global training dataset. However, not only the details of different areas in one image are different but also the same area of different images are distinct. It tends to be hard to learn global optimal γ and β. Thus, we propose to condition γ and β based on the input smoothed image I_{sm}. As a side note, our spatial feature-wise transformation is inspired by SPADE [35]. The difference part is that we learn the spatial γ and β from 3-channel smoothed images rather than predefined class labels for every pixel.

We show the spatial feature-wise transformation in Figure 2. We can observe that the input smoothed image I_{sm} is passed through

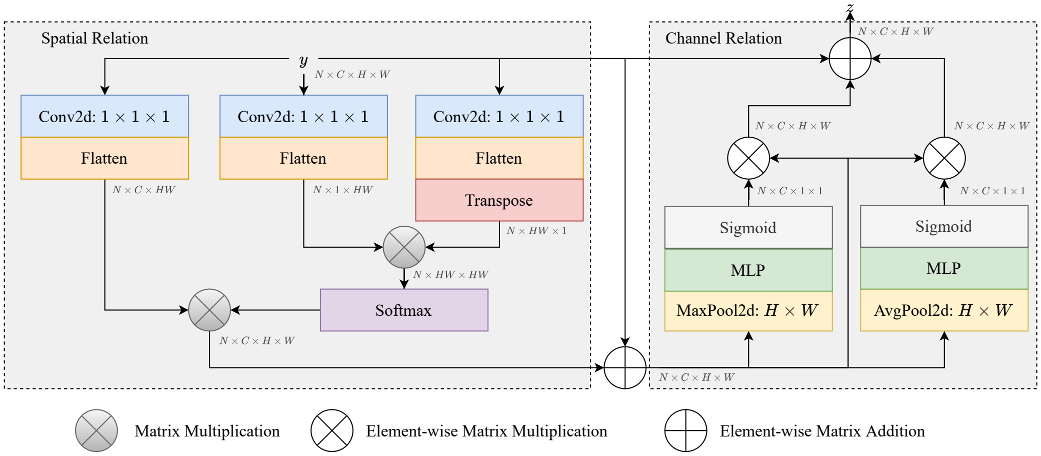


Figure 3: Full attention (FA). As for the input feature y, we consider both the channel relation and the spatial relation to generate the output feature z.

several convolutional filters to generate conditioned γ and β. The input feature x is scaled by γ and shifted by β spatially to generate feature y. Another aspect of this is that during the different upsampling phases, we interpolate the input image I_{sm} to the same height and width as the input feature.

3.2 Full Attention

Modeling relations with attention mechanisms have been utilized in several tasks [10, 42, 45, 48]. In this work, we wish to model relations between different pixels. There exist relations between the details of different pixels. For example, as for smoothed selfies in Figure 1, if we know the center pixels represent the nose, the surrounding pixels are likely to be skin. Therefore, the relations between different pixels can give us more information to recover details. However, the smoothed images are represented with multichannel features in our DeSmoothGAN. As a result, if we want to model relations between pixels, we must consider the relations between different channels. Thus, we propose full attention consisting of spatial relation and channel relation as shown in Figure 3. One key for applying attention mechanisms in a GAN is how to calculate the weights for the input feature y, as introduced below.

As for spatial relation, we employ matrix multiplication to calculate weights for each pixel. In detail, the input feature will be passed through 3 convolution layers separately as follows,

$$\begin{aligned} y_f &= f(y) = W_f * y + b_f, \\ y_g &= g(y) = W_g * y + b_g, \\ y_h &= h(y) = W_h * y + b_h, \end{aligned} \tag{2}$$

where W_f, W_g and W_h denote the 1×1 convolutional filters; b_f, b_g and b_h denote the bias. A related point to consider is that $f(y)$ and $g(y)$ are used to calculate the weights. In order to reduce heavy computation of matrix multiplication, we reduce the feature channel to 1, which means $y_f \in \mathbb{R}^{N \times 1 \times H \times W}$ and $y_g \in \mathbb{R}^{N \times 1 \times H \times W}$. In order to perform matrix multiplication, we reduce the dimension of the feature maps by flattening them. As a result, $y_f \in \mathbb{R}^{N \times 1 \times H \times W}$ becomes $y_f^{'} \in \mathbb{R}^{N \times 1 \times HW}$; $y_g \in \mathbb{R}^{N \times 1 \times H \times W}$ becomes $y_g^{'} \in \mathbb{R}^{N \times 1 \times HW}$; $y_h \in \mathbb{R}^{N \times 1 \times H \times W}$ becomes $y_h^{'} \in \mathbb{R}^{N \times 1 \times HW}$. Let M be the weight matrix for each pixel, it is calculated as follows,

$$M = y_f^{'\,T} \cdot y_g^{'}, \tag{3}$$

where T denotes the transpose operation and $\cdot$ denotes matrix multiplication. Further, we also apply a softmax layer to rescale weights to the range $[0, 1]$. Finally, we multiply the weight matrix with the feature to generate the output feature y_{sr} as follows,

$$y_{sr} = y_h^{'} \cdot M, \tag{4}$$

where $\cdot$ denotes matrix multiplication.

As for channel relation, we employ pooling operation and a multilayer perceptron to calculate weights for each channel. In detail, the input features $y^{'} = y + y_{sr}$ will be passed through a max pooling layer and an average pooling layer separately. Both max pooling layer and average pooling layer reduce $y^{'} \in \mathbb{R}^{N \times C \times H \times W}$ to $y_{pl} \in \mathbb{R}^{N \times C \times 1 \times 1}$. The max pooling layer takes the max value over the whole feature maps as follows,

$$y_{max}(N_i, C_j, 0, 0) = \max_{m=0...H-1} \Big(\max_{n=0...W-1} (y^{'}(N_i, C_j, m, n)) \Big), \tag{5}$$

while the average pooling layer takes the average value over the whole feature maps as follows,

$$y_{avg}(N_i, C_j, 0, 0) = \frac{1}{H * W} \sum_{m=0}^{H-1} \sum_{n=0}^{W-1} y^{'}(N_i, C_j, m, n), \tag{6}$$

where N_i denotes the i-th sample, C_j denotes the j-th channel, H and W denote the height and the width of the feature map. Further, we employ a multilayer perceptron and a sigmoid function to refine the weights for each channel. Then, the weights are multiplied with the corresponding channels. Finally, we sum the features from two branches including max pooling and average pooling to get the output feature z as follows,

$$z = S(MLP(y_{max})) * y^{'} + S(MLP(y_{avg})) * y^{'} + y, \tag{7}$$

where MLP denotes the same multilayer perceptron and S represents the sigmoid function.

3.3 Loss Functions

Our DeSmoothGAN consists of a generator G to recover details of smoothed images and a discriminator D to distinguish recovered images and corresponding unsmoothed images. Specifically, we put the spatial feature-wise transformation and the full attention in the upsampling phase of the generator as shown in Figure 1. We employ the *adversarial loss* [13] and *perceptual loss* [24] to train DeSmoothGAN.

Let I_{sm}, I_{re}, and I_{gt} denote the smoothed images, recovered images and corresponding unsmoothed images that are also ground truth images. The training images can be denoted as $\{I_{sm}^i\}_{i=1}^N$ and $\{I_{gt}^i\}_{i=1}^N$, where N represents the size of training images. The corresponding data distribution is denoted as $I_{sm} \sim p_{data}(I_{sm})$ and $I_{gt} \sim p_{data}(I_{gt})$.

As for *adversarial loss*, this objective can be denoted as follows,

$$\begin{aligned}\mathcal{L}_{adv}(G, D, I_{sm}, I_{gt}) = & \mathbb{E}_{I_{gt} \sim p_{data}(I_{gt})}[logD(I_{gt})] \\ & + \mathbb{E}_{I_{sm} \sim p_{data}(I_{sm})}[log(1 - D(G(I_{sm})))],\end{aligned} \tag{8}$$

where G tries to recover details of smoothed images I_{sm} and D aims to ensure recovered images I_{re} follow the same distribution of the corresponding unsmoothed image I_{gt}.

As for *perceptual loss*, this objective can be expressed as follows,

$$\mathcal{L}_{per}(G) = \sum_{i=1}^{L} w_i \parallel feat_i(I_{re}) - feat_i(I_{gt}) \parallel_1, \tag{9}$$

where $feat_i(I)$ denotes the i-th layer feature of VGG-19 [37] for the input image I, w_i represents the weight of i-th layer, and L describes the number of the layers. In this work, we use five layers including $conv1\_2$, $conv2\_2$, $conv3\_4$, $conv4\_4$, and $conv5\_4$. Their weights are 1/32, 1/16, 1/8, 1/4 and 1 respectively. We set a higher weight for higher layers of VGG-19 because we wish the recovered results keep more high-level information such as the overall structure as demonstrated in [50]. The full objective is:

$$\mathcal{L}(G, D) = \mathcal{L}_{adv}(G, D, I_{sm}, I_{gt}) + \lambda \mathcal{L}_{per}(G), \tag{10}$$

where λ controls the effect of the two different objectives. In this work, we set $\lambda = 100$ because we wish the generator G to recover as many details as possible. We aim to solve:

$$G^*, D^* = \arg \min_G \max_D \mathcal{L}(G, D), \tag{11}$$

where G^* and D^* describe the parameters of the generator G and the discriminator D, respectively.

4 EXPERIMENTS

In order to evaluate the proposed method reasonably, we first introduce the experiment settings including implementation details, datasets, and comparison baselines in Section 4.1. Then, we analyse the spatial feature transformation (SFT) and full attention (FA) in Section 4.2. Later, we propose several quantitative metrics including image similarity and smooth consistency to show objective evaluations in Section 4.3. Besides, we also conduct experiments to verify the generalization capability of DeSmoothGAN in Section 4.4.

4.1 Experiment Settings

As for the implementation, the proposed DeSmoothGAN is built on top of the general image-to-image translation framework pix2pix. Specifically, we use the batch normalization [22] for normalizing features and Adam solver [27] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for optimizing hyperparameters. We set the batch size as 4 and the learning rate as 0.0002. Furthermore, we employ the Spectral Norm [32] to the layers of the generator of DeSmoothGAN. All of the experiments in this work are conducted on a GPU of NVIDIA GeForce GTX 1080.

As for the dataset, we perform experiments on two datasets including CelebA-HQ [30] and flower [33]. Considering the goal of this work is to explore foundational techniques to recover details of smoothed images, we employ the popular smoothing technique called L_0 Smoothing [49] to generate smoothed images. As a result, we obtain 27176 training images and 2824 testing images in the CelebA-HQ dataset and 7169 training images and 1120 testing images in the flower dataset.

As for the comparison baselines, we compare three recent foundational image-to-image translation methods including pix2pix [23], AdaIN [19], and SPADE [35]. It is worth noting that both AdaIN [19] and SPADE [35] can not be directly used to perform this task of recovering details of smoothed images. Therefore, we reimplement these two ideas to achieve this task for fair comparisons. In particular, we modify SFT to generate the one-dimensional γ and β to construct AdaIN [19] by adding max pooling layer in our model. Specifically, the constructed AdaIN contains full attention (FA) module. We modify the original SPADE to learn the two-dimensional γ and β conditioned on smoothed images. Our model without full attention (FA) can also be viewed as a kind of SPADE and we name it SPADE-1 as shown in Table 1.

4.2 Analysis of SFT and FA

The goal of spatial feature-wise transformation (SFT) is to utilize the spatial variability of smoothed images and recover details of different areas differently. In order to verify its functionality, we not only display our results with and without SFT but also visualize the learned γ and β in the last SFT block in Figure 4. Specifically, we sum all the features to one dimension to visualize learned γ and β as the same as U-GAT-IT [26]. From the 2nd column in Figure 4, we can observe that although the main areas of flowers are recovered well, the top areas around the flowers are blurred and distorted. It is understandable because every area is treated equally without SFT. Therefore, our model without SFT can only learn a global optimal weight to recover most areas as much as possible. In contrast, our model with SFT can find several different optimal weights to recover different areas. Furthermore, the visualized spatial γ and β in the last two columns in Figure 4 reveal that different areas have different weights.

We propose full attention (FA) to consider the spatial correlation of smoothed images. Since images are represented as multi-channel features in current CNN-based networks, we are supposed to consider both the spatial relation and the channel relation. The spatial relation can give us more information about the missing details. For example, as shown in Figure 5, the central parts of flowers including pistils and stamens can help us inference the details of petals of flowers. Although the result without spatial relation in Figure 5 looks okay, the details of petals are different from the ground truth. The purpose of channel relations is to control the

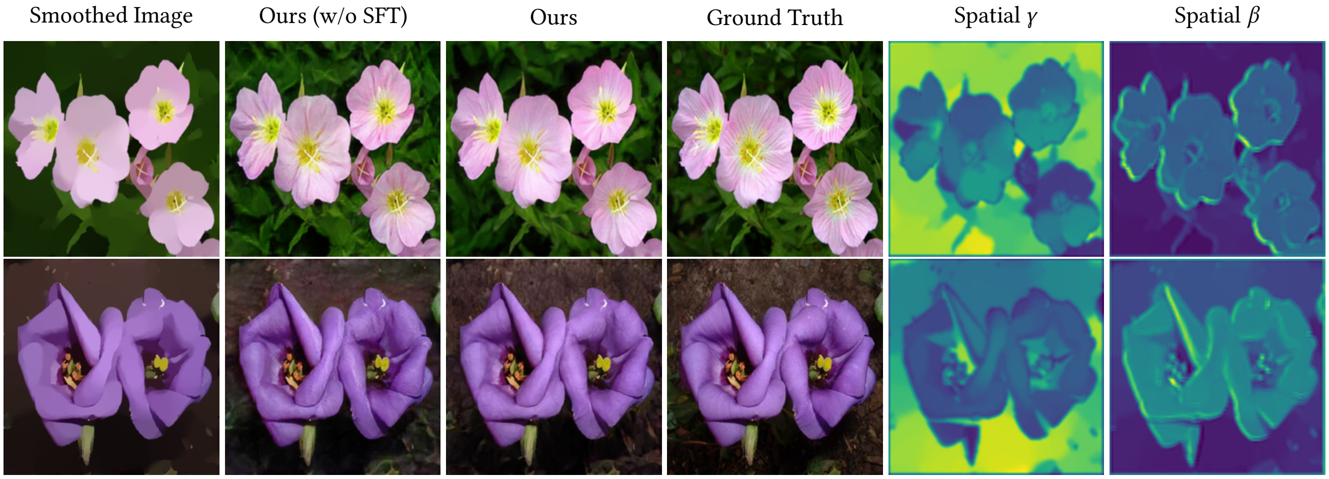
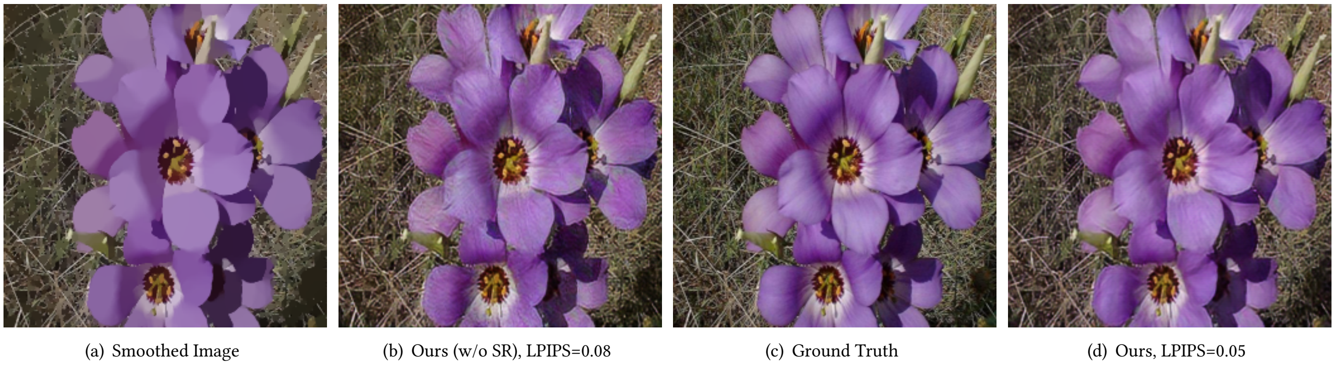


Figure 4: Effectiveness of spatial feature-wise transformation (SFT). The leftmost column shows the input smooth images. The right columns display our results without SFT, our results, ground truth images, visualized spatial γ, and visualized spatial β, respectively. We can observe that if we do not perform SFT, the top areas around flowers in the 2nd column are blurred and distorted.


(a) Smoothed Image

(b) Ours (w/o SR), LPIPS=0.08

(c) Ground Truth

(d) Ours, LPIPS=0.05

Figure 5: Effectiveness of spatial relation (SR) in full attention (FA). (a) The input smoothed image. (b) Our result without spatial relation. (c) The ground truth. (d) Our result. The details of petals in (b) do not conform to the details of (c).

weights of different channels adaptively. It can help us weaken the redundant details while enhancing the necessary details. As shown in Figure 6, the details of central petals without channel relation are overly recovered.

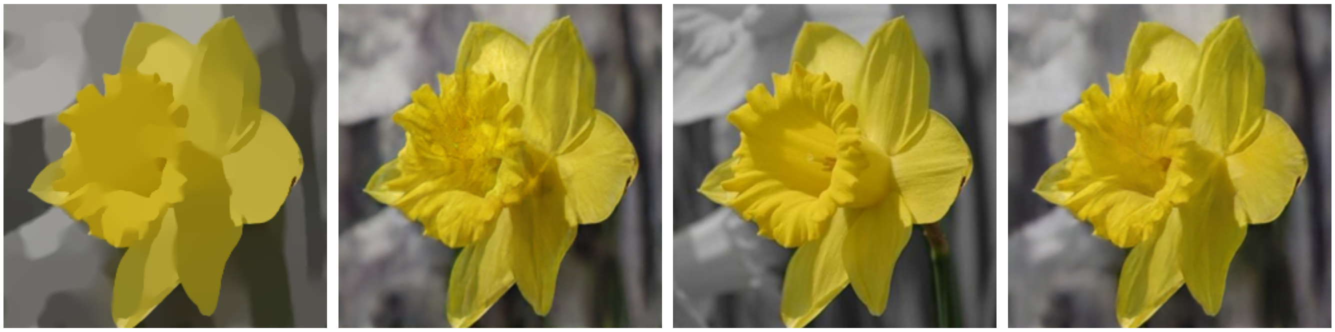
4.3 Quantitative Experiments

For smoothed images in our experiment datasets, we have their corresponding ground truth images before smoothing. Therefore, we evaluate different methods from two aspects consisting of image similarity and smooth consistency. For image similarity, it measures how similar the generated images are to the ground truth images. We not only use traditional image similarity metrics including SSIM [47] and PSNR but also employ the Learned Perceptual Image Patch Similarity (LPIPS) [54] to represent image similarity. As for smooth consistency, it means that if we apply the same smoothing algorithm to the generated images I_{re}, they should be the same as I_{sm}. We define smooth consistency as follows:

$$SC = MSE(F_{sm}(I_{re}), I_{sm}), \tag{12}$$

where F_{sm} represents the corresponding smooth algorithm and MSE denotes the mean squared error. For these different metrics, higher values of SSIM and PSNR, and lower values of SC and LPIPS, mean better results.

We show several comparisons in Figure 7. The top two samples are from the flower testing dataset and the bottom sample is from the CelebA-HQ testing dataset. From the results, we can observe that our results are either as good as others or the best among the comparisons. For instance, as for the 2nd row in Figure 7, the petal details of pix2pix [23] and SPADE [35] are overly recovered while the artifacts exist in the petals of AdaIN [19]. In contrast, compared to the ground truth, our result looks better and artifacts-free. We also display several additional recovered results in the supplementary material.



(a) Smoothed Image

(b) Ours (w/o CR), LPIPS=0.10

(c) Ground Truth

(d) Ours, LPIPS=0.08

Figure 6: Effectiveness of channel relation (CR) in full attention (FA). (a) The input smoothed image. (b) Our result without channel relation. (c) The ground truth. (d) Our result. The details of central petals in (b) are overly recovered, compared to (c).

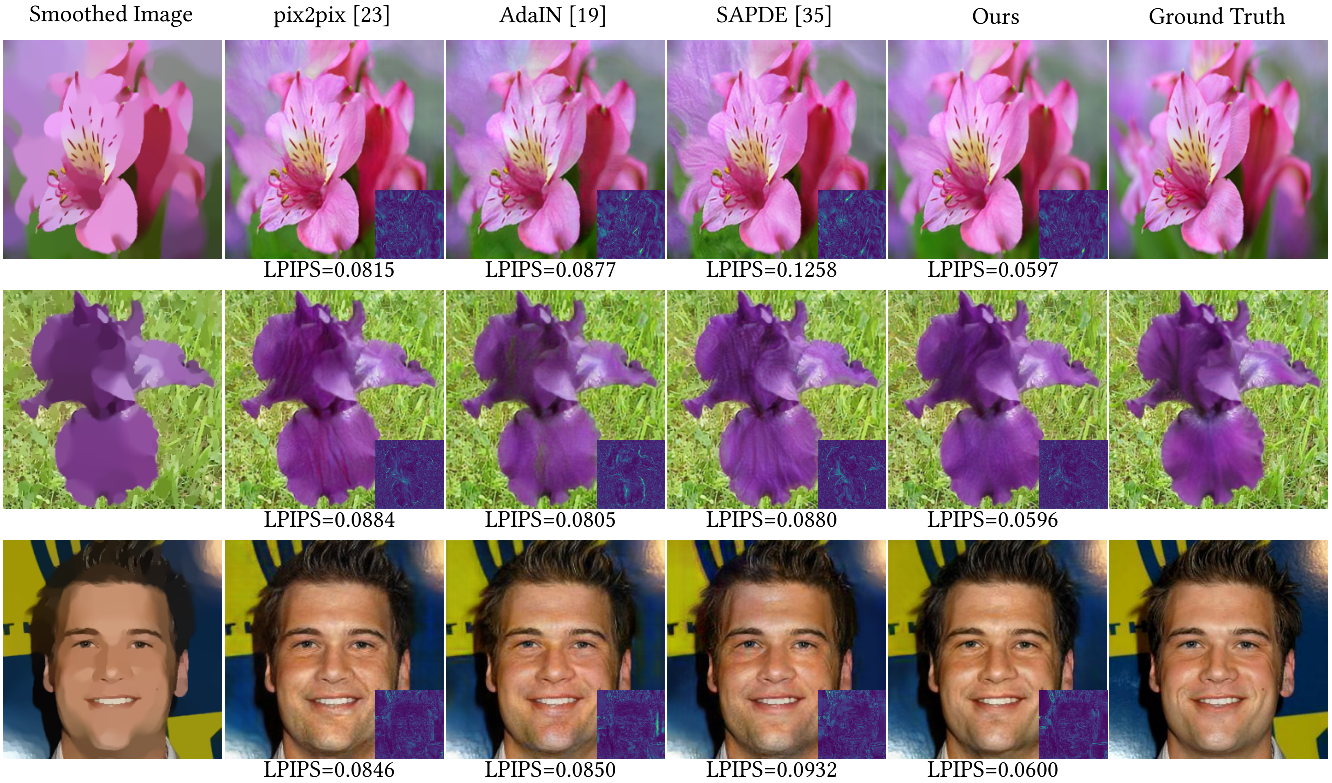


Figure 7: Comparisons with different methods. The leftmost column shows the input smooth images. The right columns display results with pix2pix [23], AdaIN [19], SPADE [35], ours, and ground truth, respectively. The better the result, the lower the LPIPS. We also display the visualized color difference map with CIEDE 2000 in the lower right corner for each comparison. The brighter the color, the greater the error.

In order to compare with different methods quantitatively, we also measure the quantitative performance including smooth consistency and image similarity with ground truth and report them in Table 1. We also display the number of parameters of different methods in Table 1. The quantitative results in the CelebA-HQ testing dataset and the flower testing dataset reveal our method achieves the best performance in SSIM, PSNR, and LPIPS. The first row and second row in Table 1 reveal the performance with a light pix2pix (16M) and a deeper pix2pix (41M) named pix2pix-EX. Simply making the network deeper while increasing the parameters of pix2pix do not generate a better performance since we ignore the spatial variability and spatial correlation of smoothed images. As for the

Table 1: Performance Comparison of Different Methods

		CelebA-HQ				Flower			
name	#param	SC↓	SSIM↑	PSNR↑	LPIPS↓	SC↓	SSIM↑	PSNR↑	LPIPS↓
pix2pix [23]	16M	37.5287	0.7414	26.7797	0.0939	46.4804	0.7224	25.4516	0.1007
pix2pix-EX	41M	34.2789	0.7470	26.9618	0.0921	33.9261	0.7296	25.9525	0.0996
AdaIN [19]	32M	32.8747	0.7618	27.2144	0.0850	42.3919	0.7414	25.8982	0.0996
SPADE [35]	56M	**31.2382**	0.7501	26.6676	0.0997	31.5496	0.7407	25.7620	0.1105
Ours	32M	32.3942	**0.7661**	**27.3601**	**0.0803**	**31.4313**	**0.7518**	**26.3831**	**0.0858**
Ours (w/o SFT)	19M	35.8944	0.7449	26.8964	0.0930	42.6541	0.7188	25.5838	0.1027
Ours (w/o FA) ≈ SPADE-1	29M	33.7224	0.7607	27.0853	0.0907	31.6741	0.7483	26.3652	0.0903
Ours (w/o SFT & FA)	16M	37.8993	0.7437	26.8243	0.0962	43.4639	0.7257	25.6171	0.1068

performance of AdaIN in Table 1, it reveals that utilizing spatial variability is important for recovering details of smoothed images since the difference between AdaIN and our method is to generate one-dimensional or two-dimensional γ and β as introduced in Section 4.1. As for the performance of SPADE and SPADE-1, both of them utilize spatial variability and generate wonderful results. However, they still ignore spatial correlation of smoothed images compared to our method. Though our performances on SC are slightly worse than SPADE on the CelebA-HQ dataset, we should not ignore the difference of the number of parameters used by the two methods: 56M (SPADE) and 32M (Ours). Overall, our method considering spatial variability and spatial correlation improves the performance of recovering details of smoothed images.

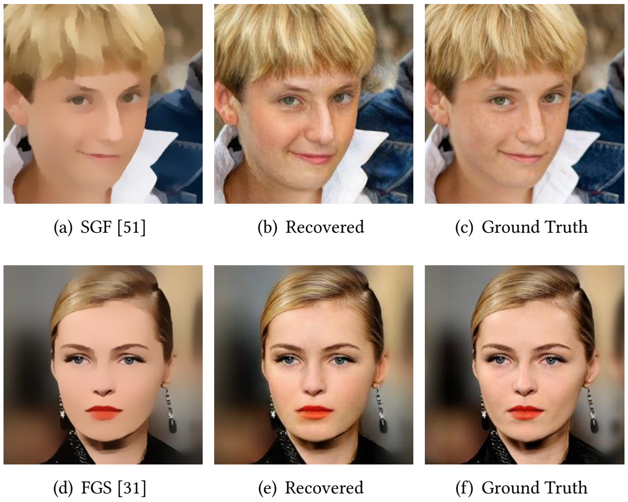
We also perform ablation studies to verify the effectiveness of spatial feature-wise transformation and full attention. We conduct three experiments including removing spatial feature-wise transformation or full attention individually and removing both of them. The quantitative results are reported in Table 1. We can observe that both spatial feature-wise transformation and full attention are beneficial to recover details of smoothed images. Specifically, if we combine them, we can achieve the best performance.

4.4 Generalization Capability

In this work, we employ the popular L_0 Smoothing technique [49] to generate training and testing datasets in order to explore the foundational technique to recover details of smoothed images. Though our DeSmoothGAN is trained on this kind of dataset, it can also be used to recover details of smoothed images generated by other smoothing algorithms. We show several results in Figure 8. We can find the recovered results of smoothing algorithms including SGF [51] and FGS [31] are close to the ground truth. However, if we want to achieve the best result on all kinds of smoothing algorithms, we would better collect enough training datasets of all kinds of smoothing algorithms, refine the network architecture, and even investigate meta-learning techniques correspondingly.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose DeSmoothGAN that utilizes the spatial feature-wise transformation considering spatial variability and the full attention considering spatial correlations to explore foundational techniques to recover details of smoothed images. However, how to perform a few-shot [38] desmoothing with limited paired training data and achieve a general image desmoothing model in the wild require further explorations. In addition, how to develop an all-in-one model to reverse the black-box beautification process including smoothing, shape deformation [44], and other operations is also an interesting avenue for the future work.



(a) SGF [51] (b) Recovered (c) Ground Truth

(d) FGS [31] (e) Recovered (f) Ground Truth

Figure 8: Generalization capability. (a) The smoothed result of SGF [51]. (b) Our recovered result of (a). (c) The ground truth of (a). (d) The smoothed result of FGS [31]. (e) Our recovered of (d). (f) The ground truth of (d).

ACKNOWLEDGMENTS

This work was partly supported by National Natural Science Foundation of China (No.61802128, 61672237, 61532002) and Hong Kong ECS grant No.21209119, Hong Kong UGC.

REFERENCES

[1] Badour AlBahar and Jia-Bin Huang. 2019. Guided image-to-image translation with bi-directional feature transformation. In *IEEE International Conference on Computer Vision (ICCV)*. 9016–9025.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

[3] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. 2019. Attention augmented convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*. 3286–3295.

[4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. 2018. Aˆ 2-nets: Double attention networks. In *Advances in Neural Information Processing Systems*. 352–361.

[5] Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron Courville. 2017. Modulating Early Visual Processing by Language. In *Advances in Neural Information Processing Systems*. 6597–6607.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*. 184–199.

[7] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. 2018. Feature-wise transformations. *Distill* 3, 7 (2018), e11.

[8] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2017. A Learned Representation For Artistic Style. In *International Conference on Learning Representations (ICLR)*.

[9] Hany Farid. 2016. *Photo forensics*. MIT Press.

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3146–3154.

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2414–2423.

[12] Manjunath Kudlur Vincent Dumoulin Golnaz Ghiasi, Honglak Lee and Jonathon Shlens. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In *British Machine Vision Conference (BMVC)*. Article 114, 12 pages.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

[14] G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[16] Xin Hong, Pengfei Xiong, Renhe Ji, and Haoqiang Fan. 2019. Deep Fusion Network for Image Completion. In *ACM International Conference on Multimedia*. 2033–2042.

[17] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. 2018. Gather-excite: Exploiting feature context in convolutional neural networks. In *Advances in Neural Information Processing Systems*. 9401–9411.

[18] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[19] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (ICCV)*. 1501–1510.

[20] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*. 172–189.

[21] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. 2019. Lightweight image super-resolution with information multi-distillation network. In *ACM International Conference on Multimedia*. 2024–2032.

[22] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*, Vol. 37. 448–456.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1125–1134.

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*. 694–711.

[25] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4401–4410.

[26] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. 2019. U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *International Conference on Learning Representations (ICLR)*.

[27] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

[28] Jiamin Li, Hui Zhang, Zaixuan Fan, Xiang He, Shimin He, Mingyuan Sun, Yimin Ma, Shiqiang Fang, Huabing Zhang, and Bingjian Zhang. 2013. Investigation of the renewed diseases on murals at Mogao Grottoes. *Heritage Science* 1, 1 (2013), 31.

[29] Yahui Liu, Marco De Nadai, Gloria Zen, Nicu Sebe, and Bruno Lepri. 2019. Gesture-to-gesture translation in the wild via category-independent conditional maps. In *ACM International Conference on Multimedia*. 1916–1924.

[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *IEEE International Conference on Computer Vision (ICCV)*.

[31] Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsub Ham, Kwanghoon Sohn, and Minh N Do. 2014. Fast global image smoothing based on weighted least squares. *IEEE Transactions on Image Processing* 23, 12 (2014), 5638–5653.

[32] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

[33] M-E Nilsback and Andrew Zisserman. 2006. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2. 1447–1454.

[34] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. 2018. BAM: Bottleneck Attention Module. In *British Machine Vision Conference (BMVC)*.

[35] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2337–2346.

[36] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[37] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*.

[38] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. 4077–4087.

[39] Florian Strub, Mathieu Seurin, Ethan Perez, Harm de Vries, Jérémie Mary, Philippe Preux, and Aaron CourvilleOlivier Pietquin. 2018. Visual reasoning with multi-hop feature modulation. In *European Conference on Computer Vision (ECCV)*. 784–800.

[40] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. 2019. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *ACM International Conference on Multimedia*. 2052–2060.

[41] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. 2016. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*. 4790–4798.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[43] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*. 2692–2700.

[44] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A Efros. 2019. Detecting photoshopped faces by scripting photoshop. In *IEEE International Conference on Computer Vision (ICCV)*. 10072–10081.

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7794–7803.

[46] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. 2018. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 606–615.

[47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[48] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. CBAM: Convolutional block attention module. In *European Conference on Computer Vision (ECCV)*. 3–19.

[49] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. 2011. Image Smoothing via L0 Gradient Minimization. *ACM Transactions on Graphics* 30, 6 (December 2011), 1–12.

[50] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*. 818–833.

[51] Feihu Zhang, Longquan Dai, Shiming Xiang, and Xiaopeng Zhang. 2015. Segment graph based image filtering: fast structure-preserving smoothing. In *IEEE International Conference on Computer Vision (ICCV)*. 361–369.

[52] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019. Self-attention generative adversarial networks. In *International Conference on Machine Learning (ICML)*, Vol. 97. 7354–7363.

[53] Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. 2018. Semantic image inpainting with progressive generative networks. In *ACM International Conference on Multimedia*. 1939–1947.

[54] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[55] Bo Zhao, Xiao Wu, Zhi-Qi Cheng, Hao Liu, Zequn Jie, and Jiashi Feng. 2018. Multi-view image generation from a single-view. In *ACM International Conference on Multimedia*. 383–391.

[56] Yuheng Zhi, Huawei Wei, and Bingbing Ni. 2018. Structure Guided Photorealistic Style Transfer. In *ACM International Conference on Multimedia*. 365–373.