Deep Texture and Structure Aware Filtering Network for Image Smoothing

Kaiyue Lu[1,2], Shaodi You[2,1], Nick Barnes[2,1]

[1] Research School of Engineering, Australian National University
[2] Data61, CSIRO
{Kaiyue.Lu, Shaodi.You, Nick.Barnes}@data61.csiro.au

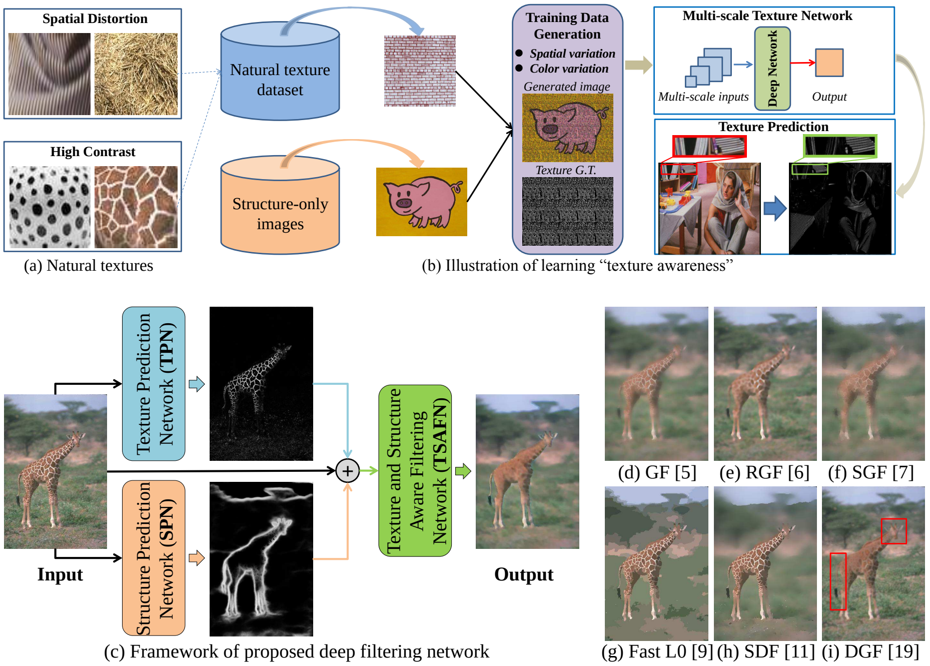
Abstract. Image smoothing is a fundamental task in computer vision, that attempts to retain salient structures and remove insignificant textures. In this paper, we aim to address the fundamental shortcomings of existing image smoothing methods, which cannot properly distinguish textures and structures with similar low-level appearance. While deep learning approaches have started to explore structure preservation through image smoothing, existing work does not yet properly address textures. To this end, we generate a large dataset by blending natural textures with clean structure-only images, and use this to build a texture prediction network (TPN) that predicts the location and magnitude of textures. We then combine the TPN with a semantic structure prediction network (SPN) so that the final texture and structure aware filtering network (TSAFN) is able to identify the textures to remove ("texture-awareness") and the structures to preserve ("structure-awareness"). The proposed model is easy to understand and implement, and shows good performance on real images in the wild as well as our generated dataset.

Keywords: Image smoothing, texture prediction, deep learning

1 Introduction

Image smoothing, a fundamental technology in image processing and computer vision, aims to clean images by retaining salient structures (to the ***structure-only image***) and removing insignificant textures (to the ***texture-only image***), with various applications including denoising [15], detail enhancement [14], image abstraction [38] and segmentation [36].

There are mainly two types of methods for image smoothing: (1) kernel-based methods, that calculate the average of the neighborhood for texture pixels while trying to retain the original value for structural pixels, such as the guided filter (GF) [18], rolling guidance filter (RGF) [46], segment graph filter (SGF) [45] and so on; and (2) separation-based methods, which decompose the image into a structure layer and a texture layer, such as relative total variation (RTV) [42], fast L0 [27], and static and dynamic guidance filter (SDF) [16,17]. Traditional approaches rely on hand-crafted features and/or prior knowledge to distinguish



(a) Natural textures

(b) Illustration of learning "texture awareness"

(c) Framework of proposed deep filtering network

(d) GF [5] (e) RGF [6] (f) SGF [7]

(g) Fast L0 [9] (h) SDF [11] (i) DGF [19]

Fig. 1. (a) Texture in natural images is often hard to identify due to spatial distortion and high contrast. (b) Illustration of learning "texture awareness". We generate training data by adding spatial and color variations to natural texture patterns and blending them with structure-only images, and then use the result to train a multi-scale texture network with texture ground-truth. We test the network on both generated data and natural images. (c) Our proposed deep filtering network is composed of a texture prediction network (TPN) for predicting textures (white stripes with high-contrast); a structure prediction network (SPN) for extracting structures (the giraffe's boundary, which has relatively low contrast to the background); and a texture and structure aware filtering network (TSAFN) for image smoothing. (d)-(i) Existing methods cannot distinguish low-contrast structures from high-contrast textures effectively.

textures from structures. These features are largely based on low-level appearance, and generally assume that structures always have larger gradients, and textures are just smaller oscillations in color intensities.

In fact, it is quite difficult to identify textures. The main reasons are twofold: (1) textures are essentially repeated patterns regularly or irregularly distributed within object structures, and they may show significant spatial distortions in an image (as shown in Fig. 1(a)), making it hard to fully define them mathematically; (2) in some images there are strong textures with large gradients and color contrast to the background, which are easy to confuse with structures (such as the white stripes on the giraffe's body in Fig. 1(c)). We see from Fig. 1 that GF, RGF, SGF, fast L0, and SDF perform poorly on the giraffe image. The textures are either not removed, or suppressed with the structure severely blurred. This is because the hand-crafted nature of these filters makes them less robust when applied to various types of textures, and also leads to poor discrimination of textures and structures. Some other methods [41,23,21,12,6,11,31] take advantage of deep neural networks, and aim for better performance by extracting richer information. However, existing networks use the output of various hand-crafted filters as ground-truth during training. These deep learning approaches are thus limited by the shortcomings of hand-crafted filters, and cannot learn how to effectively distinguish textures from structures.

A recently-proposed double-guided filter (DGF) [25] addresses this issue by introducing the idea of "texture guidance", which infers the location of texture, and combines it with "structure guidance" to achieve both goals of texture removal and structure preservation. However, DGF uses a hand-crafted separation-based algorithm called Structure Gradient and Texture Decorrelating (SGTD) [22] to construct the texture confidence map that still cannot essentially overcome the natural deficiency. We argue that this is not true "texture awareness", because in many cases, some structures are inevitably blurred when the filter tries to remove strong textures after several iterations. As can be seen in Fig. 1(i), although the stripe textures are largely smoothed out, the structure of the giraffe is unexpectedly blurred, especially around the head and the tail (red boxes).

In this paper, we hold the idea that "texture awareness" should reflect both the *texture region* (where the texture is) and *texture magnitude* (texture with high contrast to the background is harder to remove). Thus, we take advantage of deep learning and propose a texture prediction network (TPN) that aims to learn textures from natural images. However, since there are no available datasets containing natural images with labeled texture regions, we make use of texture-only datasets [8,10]. The process of learning "texture awareness" is shown in Fig. 1(b). Specifically, we generate the training data by adding spatial and color variations to natural texture patterns and blending them with the structure-only image. Then we construct a multi-scale network (containing different levels of contextual information) to train these images with texture ground-truth (G.T. in short). The proposed TPN is able to predict textures through a full consideration of both low-level appearance, *e.g.*, gradient, and other statistics, *e.g.*, repetition, tiling, spatial varying distortion. The network achieves good performance on our generated testing data, and can also generalize well to natural images, effectively locating texture regions and measuring texture magnitude by assigning different confidence, as shown in Fig. 1(b). More details can be found in Section 3.

For the full problem, we are inspired by the idea of "double guidance" introduced in [25] and propose a deep neural network based filter that learns to predict textures to remove ("texture-awareness" by our TPN) and structures to preserve ("structure-awareness" by HED semantic edge detection [39]). This is an end-to-end image smoothing architecture which we refer to as "Texture and Structure Aware Filtering Network" (TSAFN), as shown in Fig. 1(c). The network is trained with our own generated dataset. Different from the work in [25], we generate texture and structure guidance with deep learning approaches, and replace the hand-crafted kernel filter with a deep learning model to achieve a more consistent and effective combination of these two types of guidance. Experimental results show that our proposed filter outperforms DGF [25] in terms of both effectiveness and efficiency, achieves state-of-the-art performance on our dataset, and generalizes well to natural images.

The main contributions of this paper are: (1) We propose a deep neural network to robustly predict textures in natural images. (2) We present a large dataset that enables training texture prediction and image smoothing. (3) We propose an end-to-end deep neural network for image smoothing that achieves both "texture-awareness" and "structure-awareness", and outperforms existing methods on challenging natural images.

2 Related Work

Texture extraction from structures The basic assumption of this type of work is that an image can be decomposed into structure and texture layers (the structure layer is a smoothed version of the input and contains salient structures, while the texture layer contains insignificant details or textures). The pioneering work, Total Variation [30], aims to minimize the quadratic difference between the input and output images to maintain structure consistency with the gradient loss as an additional penalty. Later works retain the quadratic form and propose other regularizer terms or features (*gradient loss is still necessary to keep the structures as sharp as possible*), such as weighted least squares (WLS) [13], ℓ_0 norm smoothing [40,27], ℓ_1 norm smoothing [3], local extrema [32], structure gradient and texture decorrelating (SGTD) [22]. Other works also focuses on accelerating the optimization [4] or improving existing algorithms [24]. There are two general issues that have not been handled effectively in existing work. Firstly, as they are largely dependent on gradient information, these methods *lack discrimination of textures and structures*, especially when they have similar low-level appearance, particularly in terms of scale or magnitude. Secondly, all the objective functions are *manually defined*, and may not be adaptive and robust to the huge variety of possible textures, especially in natural images.

Image smoothing with guidance The guidance image can provide structure information to help repair and sharpen structures in the target image. Since adding guidance into separation-based methods may make it harder to optimize, this idea is more widely used in kernel-based methods. Static guidance refers to the use of a fixed guidance image, such as the bilateral filter [35], joint bilateral filter [28], and guided filter [18]. To make the guidance more structure-aware, existing filters also employ techniques such as leverage tree distance [2], superpixels [45], region covariances [20], co-occurrence matrix [19], propagation distance [29], multipoint estimation [34], fully connected regions [9] and edge maps [43,7,44]. In contrast, dynamic guidance methods update the guidance image to suppress more details [46,16,17] by iteratively refining the target image. Overall, the aforementioned guidance methods only address structure information, or assume that structures and textures can be sufficiently distinguished with a single guidance. However, in most cases, structures and textures interfere with each other severely. Lu *et al.* [25] address this issue by introducing the concept of "texture guidance", which infers texture regions by normalizing the texture layer separated by SGTD [22] to construct the texture confidence map. They then naively combine it with structure guidance to form a double-guided kernel filter. However, this method is still largely dependent on hand-crafted features (in particular it relies on the hand-crafted SGTD to infer textures, which is not robust in essence). Structures may be blurred when the filter tries to smooth out strong textures after several iterations.

Deep image smoothing Deep learning has been widely used in low-level vision tasks, such as super resolution [33], deblurring [26] and dehazing [5]. Compared with non-learning approaches, deep learning is able to extract richer information from images. In image smoothing, current deep filtering models all focus on approximating and accelerating existing non-learning filters. [41] is a pioneering paper, where the learning is performed on the gradient domain and the output is reconstructed from the refined gradients produced by the deep network. Liu *et al.* [23] take advantage of both convolutional networks (for perceiving salient structures) and recurrent networks (for producing smoothing output in a data-driven manner). Li *et al.* [21] fuse the features from the original input and guidance image together and then produce the guided smoothing result (this work is mainly for upsampling). Fan *et al.* [12] first construct a network called E-CNN to predict the edge/structure confidence map based on gradients, and then use it to guide the filtering network called I-CNN. Similar work can be found in [11] by the same authors. Most recent works mainly focus on extracting richer information from input images ([31] introduces a convolutional neural pyramid to extract features of different scales, and [6] utilizes context aggregation networks to include more contextual information) and yielding more satisfying results. One common issue is all of these approaches have to take the output of existing filters as ground-truth. Hence, they are unable to overcome their deficiency in discriminating textures.

3 Texture Prediction

In this section, we give insights into textures in natural images, which inspire the design of the texture prediction network (TPN) and the dataset for training.

3.1 What is texture?

Appearance of texture It is well known that many different types of textures occur in nature and it is difficult to fully define them mathematically. Generally speaking, textures are repeated patterns regularly or irregularly distributed within object structures. For example, in Fig. 1(c), the white stripes on the giraffe's surface are recognized as textures. In Fig. 2, textures are widely spread in the image on clothes, books, and the table cloth. For cognition and vision tasks,

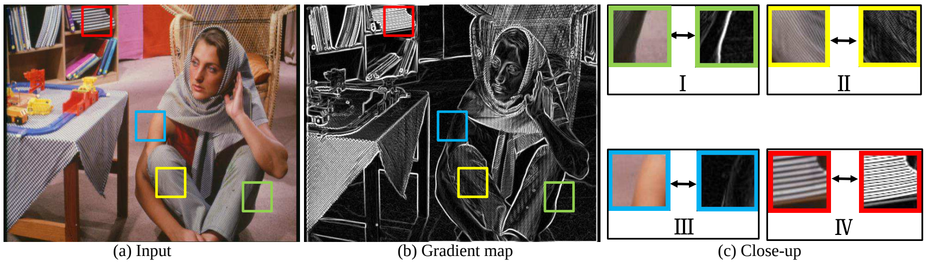


Fig. 2. Close observation of structures and textures. In contrast with the assumptions used in existing methods, large gradients do not necessarily indicate structures (IV), and small gradients may also belong to structures (III). The challenge to distinguish them motivates us to propose two independent texture and structure guidance.

an intuitive observation is that the removal of these textures will not affect the spatial structure of objects. Thus, they can be removed by image smoothing as a preprocessing step for other visual tasks.

Textures do not necessarily have small gradients Existing methods generally assume that textures are minor oscillations and have small gradients. Thus, they can easily hand-craft the filter or loss function. However, in many cases, textures may also have large gradients, *e.g.*, the white stripes on the giraffe's body in Fig. 1(b), and the stripes occurring on the books in close-up IV of Fig. 2(c). Therefore, defining textures purely based on local contrast is insufficient.

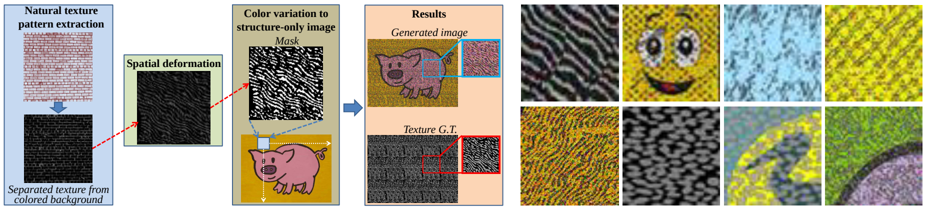
Mathematically modeling texture repetition is non-trivial By definition, textures are patterns with spatial repetitions. However, modeling and describing the repetition is non-trivial due to the existence of various distortions (see Fig. 1(a)).

Learn to predict textures To tackle these issues, we take advantage of deep neural networks. Provided sufficient training examples are available, the network is able to learn to predict textures without explicit modeling.

3.2 Dataset Generation

We aim to provide a dataset so that a deep network can learn to predict textures. Ideally, we would like to learn directly from natural images. However, manually annotating pixel-wise labels plus alpha-matting would be costly. Moreover, it would require a full range of textures, each with a full range of distortions in a broad array of natural scenes. Therefore, we propose a strategy to generate the training and testing data. Later, we will demonstrate that the proposed network is able to predict textures in the wild effectively.

We observe that cartoon images have only structural edges filled with pure color, and can be safely considered "structure-only images". Specifically, we s-elect 174 cartoon images from the Internet and 233 different types of natural


Natural texture pattern extraction
Separated texture from colored background
Spatial deformation
Color variation to structure-only image
Mask
Results
Generated image
Texture G.T.

(a) Data generation process

(b) More examples of generated data

Fig. 3. Illustration of dataset generation. We blend-in natural texture patterns to structure-only images, adding spatial and color variation to increase texture diversity. We mainly focus on image patches motivated by the fact that textures are always clustered in certain regions.

texture-only images from public datasets [8,10]. The data generation process is illustrated in Fig. 3(a). Note that texture images in these datasets show textures only and all have simple backgrounds, so that separating them from the colored background is simple and efficient even using Relative Total Variation (RTV) [42]. The texture layer separated by RTV is normalized to $[0, 1]$.

Texture itself can be irregular, and textures in the wild may be distorted because of geometric projection. This arises because textures can appear on planar surfaces that are not orthogonal to the viewing direction, as well as being projected onto object with complex 3D surfaces. Therefore, we apply both spatial and color variation to the regular textures during dataset generation. As shown in Fig. 3(a), we blend-in the texture to the structure-only image. In detail, we rescale all the texture images to 100×100 and extract texture patterns with RTV. We model spatial variation, capturing projected texture at patch level by performing geometric transforms including rotation, scaling, shearing, and linear and non-linear distortion. We randomly select the geometric transform and parameters for the operation. Based on the deformed result, we generate a binary mask $\mathbf{M}$.

As for color variation, given the structure-only image $\mathbf{S}$, the value of pixel i in the j^{th} channel of the generated image $\mathbf{I}_i^{(j)}$ is determined by both $\mathbf{S}$ and the mask $\mathbf{M}$. If $\mathbf{M}_i = 1$, $\mathbf{I}_i^{(j)} = rand[\kappa \cdot (1 - \mathbf{S}_i^{(j)}), 1 - \mathbf{S}_i^{(j)}]$, where κ is used to control the range of random generation and empirically set as 0.75. Otherwise, $\mathbf{I}_i^{(j)} = \mathbf{S}_i^{(j)}$. We repeat this by sliding the mask over the whole image without overlapping. The ground-truth texture confidence is calculated by averaging the values of the three channels of the texture layer:

$$\mathbf{T}_i^* = \delta(\frac{1}{3}\sum_{j=1}^{3}\left|\mathbf{I}_i^{(j)} - \mathbf{S}_i^{(j)}\right|), \tag{1}$$

where $\delta(x) = 1/(1 + exp(-x))$ is the sigmoid function to scale the value to $[0, 1]$. We use this color variation to generate significant contrast between the textures and the background. Using this method, it is unlikely that two images have

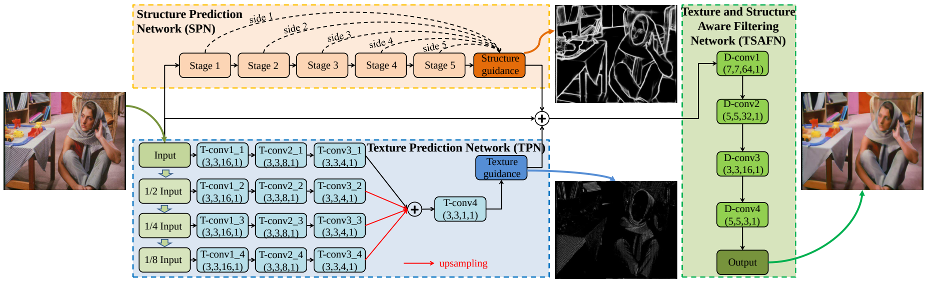


Fig. 4. Our proposed network architecture. The outputs of the texture prediction network (TPN) and structure prediction network (SPN) are concatenated with the original input, and then fed to the texture and structure aware filtering network (TSAFN) to produce the final smoothing result. (k,k,c,s) for a convolutional layer means the kernel is $k \times k$ in size with c feature maps, and the stride is s.

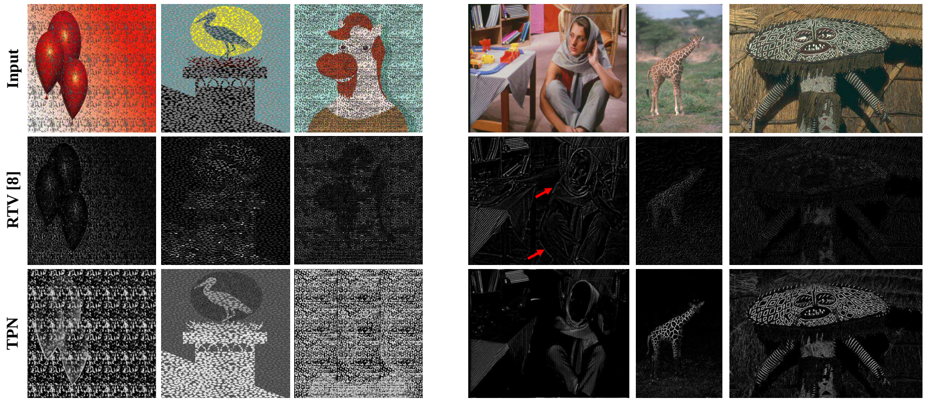
the same textures even when the textures come from the same original pattern. Fig. 3(b) shows eight generated image patches.

Finally, we generate 30,000 images in total (a handful of low-quality images have been manually removed). For ground-truth, besides the purely-clean structure-only images, we also provide binary structure maps and texture confidence maps of all the generated images. Currently we distribute textures over the entire image and the textures are not object-dependent, which may be typical appearance in natural images. Later we will show the use of patch learning can bridge the gap, motivated by the fact that textures are always clustered in certain regions. Moreover, we aim to let the network learn the statistics of appearance of local textures rather than the global structure.

3.3 Texture prediction network

Network design We propose the texture prediction network (TPN), which is trained on our generated dataset. Considering that textures have various colors, scales, and shapes, we employ a multi-scale learning strategy. Specifically, we apply 1/2, 1/4, and 1/8 down-sampling to the input respectively. For each image, we use 3 convolutional layers for feature extraction, with the same size 3×3 kernel and different number of feature maps. Then, all the feature maps are resized to the original input size and concatenated to form a 16-channel feature map. They are further convolved with a 3×3 layer to yield the final 1-channel result. Note that each convolutional layer is followed by ReLU except for the output layer, which is followed by a sigmoid activation function to scale the values to $[0, 1]$. The architecture of TPN is shown in Fig. 4. Consequently, given the input image $\mathbf{I}$, the predicted texture guidance $\tilde{\mathbf{T}}$ is obtained by:

$$\tilde{\mathbf{T}} = g\left(\mathbf{I}, \frac{1}{2}\mathbf{I}, \frac{1}{4}\mathbf{I}, \frac{1}{8}\mathbf{I}\right). \tag{2}$$



(a) Texture prediction results on generated images

(b) Texture prediction results on natural images

Fig. 5. Texture prediction results. First row: input (including both generated and natural images). Second row: texture extraction results by RTV [42] (we compare it because we use it to extract textures from texture-only images). Third row: texture prediction results by our proposed TPN. The network is able to find textures in both generated and natural images effectively, and indicate the magnitude of textures by assigning pixel-level confidence.

Network training The network is trained by minimizing the mean squared error (MSE) between the predicted texture guidance map and the ground-truth:

$$\ell_T(\theta) = \frac{1}{N}\sum_i \left\|\tilde{\mathbf{T}}_i - \mathbf{T}_i^*\right\|_2^2, \tag{3}$$

where N is the number of pixels in the image, $*$ denotes the ground-truth, and θ represents parameters. More training details can be found in the experiment section.

Texture prediction results We present the texture prediction results on our generated images in Fig. 5(a) and natural images in Fig. 5(b). The network is able to find textures in both the generated and natural images effectively, and indicate the magnitude of textures by assigning pixel-level confidence (the third row). For comparison, we also list the texture extraction results from these examples by RTV [42] in the second row. RTV performs worse on the more complex scenes, because just like other hand-crafted filters, RTV also assumes structures have large gradients and hence has poor discrimination of strong textures.

3.4 Texture and structure aware filtering network

Network design Once the structure and texture guidance are generated, the texture and structure aware filtering network (TSAFN) concatenates them with the input to form a 5-channel tensor. TSAFN consists of 4 layers. We set a relatively large kernel (7×7) in the first layer to take more original information into account. The kernel size decreases in the following two layers (5×5, 3×3 respectively). In the last layer, the kernel size is increased to 5×5 again. The first three layers are followed by ReLU, while the last layer has no activation function (transforming the tensor into the 3-channel output). Empirically, we remove all the pooling layers, the same as [41,21,12,6]. We set the filtering network without any guidance as the baseline. The whole process can be denoted as:

$$\tilde{\mathbf{I}} = h(\mathbf{I}, \tilde{\mathbf{E}}, \tilde{\mathbf{T}}). \tag{4}$$

Network training The network is trained by minimizing:

$$\ell_D(\theta) = \frac{1}{N} \sum_i (\left\| \tilde{\mathbf{I}}_i - \mathbf{I}_i^* \right\|_2^2). \tag{5}$$

More details can be found in the experiment section.

4 Experiments and Analysis

In this section, we demonstrate the effectiveness of our proposed deep image smoothing network through.

Environment setup We construct the networks in Tensorflow [1], and train and test all the data on a single NVIDIA Titan X graphics card.

Dataset Because there is no publicly-available texture removal dataset, we perform training using our generated images. More specifically, we select 19,505 images (65%) from the dataset for training, 2,998 (10%) for validation, and 7,497 (25%) for testing (all test images are resized to 512×512). There is no overlapping of structure-only and texture images between training, validation and testing samples.

Training We first train the three networks separately. 300,000 patches with the size 64×64 are randomly and sparsely collected from training images. We use gradient descent with a learning rate of 0.0001, and momentum of 0.9. Finally, we perform fine-tuning by jointly training the whole network with a smaller learning rate of 0.00001, and the same momentum 0.9. The fine-tuning loss is

$$\ell(\theta) = \gamma \cdot \ell_D(\theta) + \lambda \cdot (\ell_T(\theta) + \ell_E(\theta)), \tag{6}$$

where we empirically set $\gamma = 0.6$, and $\lambda = 0.2$.

4.1 Existing methods to compare

Non-learning methods We compare our filter with 2 classical filters: Total Variation (TV) [30], bilateral filter (BLF) [35], and 9 recently-proposed filters: L0 [40], Relative Total Variation (RTV) [42], guided filter (GF) [18], Structure

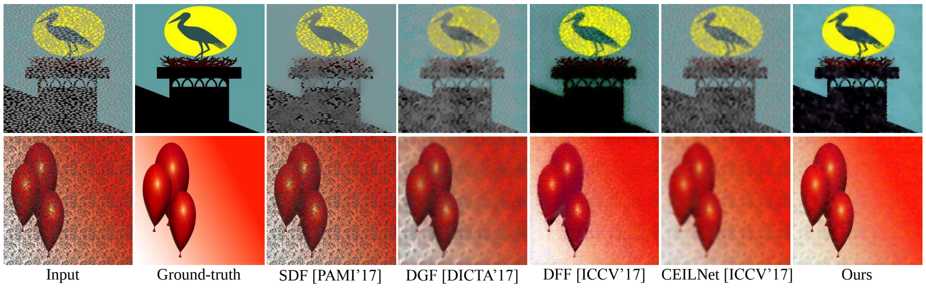


Fig. 6. Smoothing results on generated images. Our filter can smooth out various types of textures while preserving structures more effectively than other approaches.

Gradient and Texture Decorrelation (SGTD) [22], rolling guidance filter (RGF) [46], fast L0 [27], segment graph filter (SGF) [45], static and dynamic filter (SDF) [17], double-guided filter (DGF) [25]. Note that, BLF, GF, RGF, SGF, DGF are kernel-based, while TV, L0, RTV, SGTD, fast L0, SDF are separation-based. We use the default parameters defined in the open-source code for each method.

Deep learning methods We select 5 state-of-the-art deep filtering models: deep edge-aware filter (DEAF) [41], deep joint filter (DJF) [21], deep recursive filter (DRF) [23], deep fast filter (DFF) [6], and cascaded edge and image learning network (CEILNet) [12] . **We retrain all the models with our dataset**.

4.2 Results

Quantitative results on generated images We first compare the average MSE, PSNR, SSIM [37], and processing time (in seconds) of 11 non-learning filters on our testing data in Table 1. Our method achieves the smallest MSE (closest to ground-truth), largest PSNR and SSIM (removing textures and preserving main structures most effectively), and lowest running time, indicating its superiority in both effectiveness and efficiency. Note that although the double-guided filter (DGF) [25] achieves better quantitative results than other hand-crafted approaches, it runs quite slowly (more than *50 times* slower than ours). We also compare the quantitative results on different deep models trained and tested on our dataset in Table 2. Our model achieves the best MSE, PSNR and SSIM, with comparable efficiency to the other methods. We additionally select 4 state-of-the-art methods (SDF [17], DGF [25], DFF [6], and CEILNet [12]) for visual comparison in Fig. 6. Our method gives favorable qualitative results.

Qualitative comparison on real images in the wild We visually compare smoothing results of 5 challenging natural images with SDF [17], DGF [25], DF-F [6], and CEILNet [12] in Fig. 7. In the first example, the leopard is covered with black texture, and it has relatively low contrast to the background (weak structure). Only our filter smooths out most of the textures while effectively

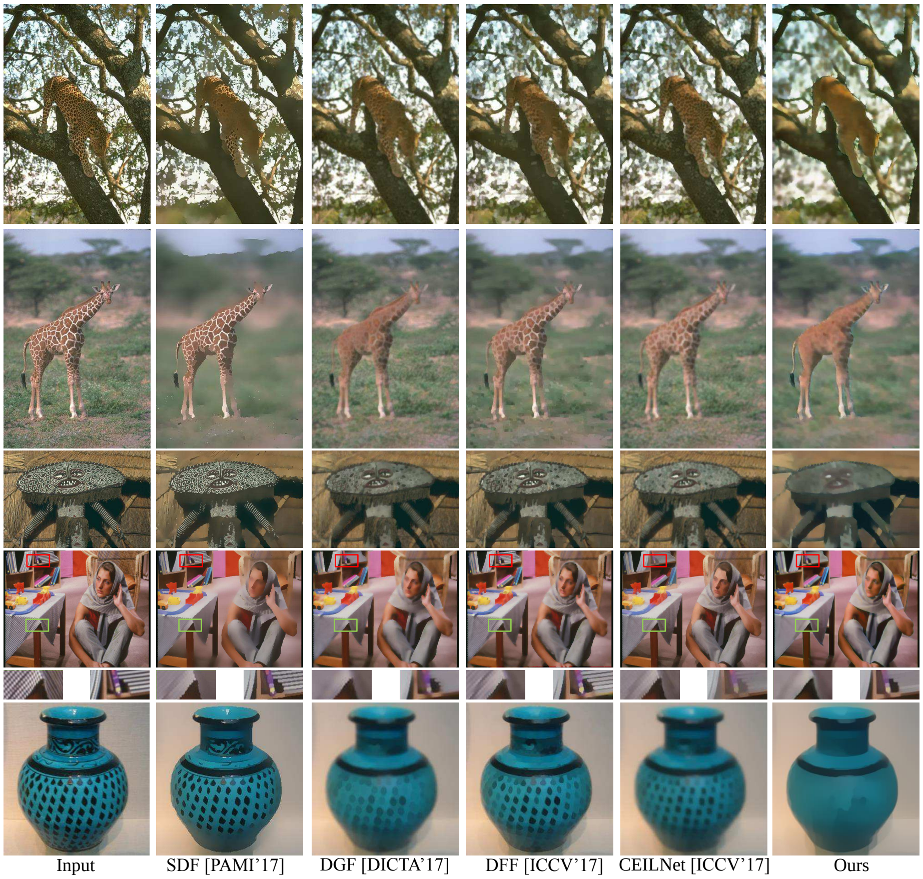


Fig. 7. Smoothing results on natural images. The first example shows the ability of weak structure preservation and enhancement in textured scenes. The next four examples present various texture types with different shapes, contrast, and distortion. Our filter performs consistently better than state-of-the-art methods in all the examples.

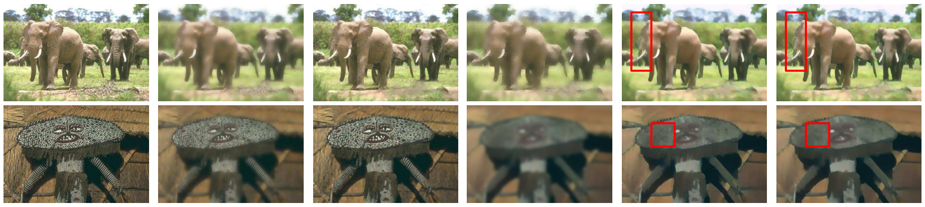
preserving and enhancing the structure. The next four examples present various texture types with different shapes, contrast, and distortion, and our filter performs consistently well. We analyze the last challenging vase example in more detail. The vase is covered with strong dotted textures, densely wrapped on the surface. SDF fails to remove these textures since they are regarded as structures with large gradients. DGF smooths out the black dots more effectively but the entire image looks blurry. This is because just as [25] points out, a larger kernel size and more iterations are required to remove more textures, resulting in the blurred structure as a penalty. Also, the naive combination of structure and texture kernels makes the filter not robust to various types of textures. The two deep filters do not demonstrate much improvement over the hand-crafted approaches

Table 1. Quantitative evaluation of different non-learning filters tested on our dataset

	MSE	PSNR	SSIM	Time		MSE	PSNR	SSIM	Time
TV [30]	0.2791	11.33	0.6817	2.44	RGF [46]	0.2094	15.73	0.7173	0.87
BLF [35]	0.3131	10.89	0.6109	4.31	Fast L0 [27]	0.2068	15.50	0.7359	1.36
L0 [40]	0.2271	14.62	0.7133	0.94	SGF [45]	0.2446	13.92	0.7002	2.26
RTV [42]	0.2388	14.07	0.7239	1.23	SDF [17]	0.1665	16.82	0.7633	3.71
GF [18]	0.2557	12.22	0.6948	0.83	DGF [25]	0.1247	17.89	0.7552	8.66
SGTD [22]	0.1951	16.14	0.7538	1.59	Ours	**0.0051**	**25.07**	**0.9152**	**0.16**

Table 2. Quantitative evaluation of deep models trained and tested on our dataset

	MSE	PSNR	SSIM	Time		MSE	PSNR	SSIM	Time
DEAF [41]	0.0297	20.62	0.8071	0.35	DFF [6]	0.0172	22.21	0.8675	**0.07**
DJF [21]	0.0352	19.01	0.7884	0.28	CEILNet [12]	0.0156	22.65	0.8712	0.13
DRF [23]	0.0285	21.14	0.8263	0.12	Ours	**0.0051**	**25.07**	**0.9152**	0.16



Input

No guidance

Structure guidance only

Texture guidance only

Double guidance (trained separately)

Double guidance (fine-tuned)

Fig. 8. Image smoothing results with no guidance, single guidance, double guidance (trained separately, and fine-tuned). With only structure guidance, the main structures are retained as well as the textures. With only texture guidance, all the textures are smoothed out but the structures are severely blurred. The result with double guidance performs well in both structure preservation and texture removal. Fine-tuning the whole network can further improve the performance.

because "texture-awareness" is not specially emphasized in their network design. Only our filter removes textures without blurring the main structure.

Ablation study of each guidance To investigate the effect of guidance, we train the filtering network with no guidance, only structure guidance, only texture guidance, and double guidance respectively. We list the average MSE, PSNR, and SSIM of the testing results compared with ground-truth in Table 3, demonstrating that the results with double guidance have smaller MSE, larg-

Table 3. Ablation study of image smoothing effects with no guidance, only structure guidance, only texture guidance, and double guidance (trained separately and fine-tuned)

	MSE	PSNR	SSIM
No guidance (Baseline)	0.0316	20.32	0.7934
Only structure guidance	0.0215	21.71	0.8671
Only texture guidance	0.0118	23.23	0.8201
Double guidance (trained separately)	0.0059	24.78	0.9078
Double guidance (fine-tuned)	**0.0051**	**25.07**	**0.9152**

er PSNR, and larger SSIM. Also, the fine-tuning process improves the filtering network. Further, we show two natural images in Fig. 8. Compared with the baseline without guidance, the result only with structure guidance retains more structure, as well as the texture (this is mainly because HED may also be negatively affected by strong textures, resulting in a larger MSE loss when training the network). In contrast, the structures are severely blurred with only texture guidance, even though most textures are removed. Combining both structure and texture guidance produces a better result due to the good discrimination of structures and textures. Fine-tuning further improves the result (in the red rectangle of the first example, the structures are sharper; in the second example, the textures within the red region are further suppressed) because it takes all the three networks as a whole and this synergistic strategy allows different features to support and complement each other for better performance. All the observations are consistent with the quantitative evaluation in Table 3.

5 Conclusion

In this paper, we propose an end-to-end texture and structure aware filtering network that is able to smooth images with both "texture-awareness" and "structure-awareness". The "texture-awareness" benefits from the newly-proposed texture prediction network. To facilitate training, we blend-in natural textures onto structure-only cartoon images with spatial and color variations. The "structure-awareness" is realized by semantic edge detection. Experiments show that the texture network can predict textures effectively. And our filtering network outperforms other filters on both generated images and natural images. The network structure is intuitive and easy to implement. In future work, we will introduce object-based texture cues - moving to a more object-based approach for texture prediction and image smoothing.

Acknowledgements This work has been partially funded by NHMRC (National Health and Medical Research Council) under Grant No.1082358.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
2. Bao, L., Song, Y., Yang, Q., Yuan, H., Wang, G.: Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. IEEE Transactions on Image Processing **23**(2), 555–569 (2014)
3. Bi, S., Han, X., Yu, Y.: An l 1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. ACM Transactions on Graphics (TOG) **34**(4), 78 (2015)
4. Buades, A., Le, T.M., Morel, J.M., Vese, L.A.: Fast cartoon+ texture image filters. IEEE Transactions on Image Processing **19**(8), 1978–1986 (2010)
5. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: An end-to-end system for single image haze removal. IEEE Transactions on Image Processing **25**(11), 5187–5198 (2016)
6. Chen, Q., Xu, J., Koltun, V.: Fast image processing with fully-convolutional networks. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
7. Cho, H., Lee, H., Kang, H., Lee, S.: Bilateral texture filtering. ACM Transactions on Graphics (TOG) **33**(4), 128 (2014)
8. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2014)
9. Dai, L., Yuan, M., Zhang, F., Zhang, X.: Fully connected guided image filtering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 352–360 (2015)
10. Dana, K.J., Van Ginneken, B., Nayar, S.K., Koenderink, J.J.: Reflectance and texture of real-world surfaces. ACM Transactions On Graphics (TOG) **18**(1), 1–34 (1999)
11. Fan, Q., Wipf, D.P., Hua, G., Chen, B.: Revisiting deep image smoothing and intrinsic image decomposition. CoRR **abs/1701.02965** (2017), http://arxiv.org/abs/1701.02965
12. Fan, Q., Yang, J., Hua, G., Chen, B., Wipf, D.: A generic deep architecture for single image reflection removal and image smoothing. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
13. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In: ACM Transactions on Graphics (TOG). vol. 27, p. 67. ACM (2008)
14. Fattal, R., Agrawala, M., Rusinkiewicz, S.: Multiscale shape and detail enhancement from multi-light image collections. ACM Trans. Graph. **26**(3), 51 (2007)
15. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2862–2869 (2014)
16. Ham, B., Cho, M., Ponce, J.: Robust image filtering using joint static and dynamic guidance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4823–4831 (2015)
17. Ham, B., Cho, M., Ponce, J.: Robust guided image filtering using nonconvex potentials. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)

18. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE transactions on pattern analysis and machine intelligence **35**(6), 1397–1409 (2013)
19. Jevnisek, R.J., Avidan, S.: Co-occurrence filter. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
20. Karacan, L., Erdem, E., Erdem, A.: Structure-preserving image smoothing via region covariances. ACM Transactions on Graphics (TOG) **32**(6), 176 (2013)
21. Li, Y., Huang, J.B., Ahuja, N., Yang, M.H.: Deep joint image filtering. In: European Conference on Computer Vision. pp. 154–169. Springer (2016)
22. Liu, Q., Liu, J., Dong, P., Liang, D.: Sgtd: Structure gradient and texture decorrelating regularization for image decomposition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1081–1088 (2013)
23. Liu, S., Pan, J., Yang, M.H.: Learning recursive filters for low-level vision via a hybrid neural network. In: European Conference on Computer Vision. pp. 560–576. Springer (2016)
24. Liu, W., Chen, X., Shen, C., Liu, Z., Yang, J.: Semi-global weighted least squares in image filtering. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
25. Lu, K., You, S., Barnes, N.: Double-guided filtering: Image smoothing with structure and texture guidance. In: The IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA) (Dec 2017)
26. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
27. Nguyen, R.M., Brown, M.S.: Fast and effective l0 gradient minimization by region fusion. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 208–216 (2015)
28. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM transactions on graphics (TOG) **23**(3), 664–672 (2004)
29. Rick Chang, J.H., Frank Wang, Y.C.: Propagated image filtering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10–18 (2015)
30. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1-4), 259–268 (1992)
31. Shen, X., Chen, Y., Tao, X., Jia, J.: Convolutional neural pyramid for image processing. CoRR **abs/1704.02071** (2017), http://arxiv.org/abs/1704.02071
32. Subr, K., Soler, C., Durand, F.: Edge-preserving multiscale image decomposition based on local extrema. ACM Transactions on Graphics (TOG) **28**(5), 147 (2009)
33. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
34. Tan, X., Sun, C., Pham, T.D.: Multipoint filtering with local polynomial approximation and range guidance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2941–2948 (2014)
35. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Computer Vision, 1998. Sixth International Conference on. pp. 839–846. IEEE (1998)
36. Wang, Y., He, C.: Image segmentation algorithm by piecewise smooth approximation. EURASIP Journal on Image and Video Processing **2012**(1), 16 (2012)
37. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

38. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. In: ACM Transactions On Graphics (TOG). vol. 25, pp. 1221–1226. ACM (2006)
39. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1395–1403 (2015)
40. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l 0 gradient minimization. In: ACM Transactions on Graphics (TOG). vol. 30, p. 174. ACM (2011)
41. Xu, L., Ren, J., Yan, Q., Liao, R., Jia, J.: Deep edge-aware filters. In: Proceedings of the 32nd International Conference on Machine Learning (ICML-15). pp. 1669–1678 (2015)
42. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Transactions on Graphics (TOG) **31**(6), 139 (2012)
43. Yang, Q.: Semantic filtering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4517–4526 (2016)
44. Zang, Y., Huang, H., Zhang, L.: Guided adaptive image smoothing via directional anisotropic structure measurement. IEEE transactions on visualization and computer graphics **21**(9), 1015–1027 (2015)
45. Zhang, F., Dai, L., Xiang, S., Zhang, X.: Segment graph based image filtering: Fast structure-preserving smoothing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 361–369 (2015)
46. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: European Conference on Computer Vision. pp. 815–830. Springer (2014)