This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

A Benchmark for Edge-Preserving Image Smoothing

Feida Zhu, *Student Member, IEEE,* Zhetong Liang, *Student Member, IEEE,* Xixi Jia, *Student Member, IEEE,* Lei Zhang, *Fellow, IEEE,* and Yizhou Yu, *Fellow, IEEE*

***Abstract*—Edge-preserving image smoothing is an important step for many low-level vision problems. Though many algorithms have been proposed, there are several difficulties hindering its further development. First, most existing algorithms cannot perform well on a wide range of image contents using a single parameter setting. Second, the performance evaluation of edge-preserving image smoothing remains subjective, and there lacks a widely accepted datasets to objectively compare the different algorithms. To address these issues and further advance the state of the art, in this work we propose a benchmark for edge-preserving image smoothing. This benchmark includes an image dataset with groundtruth image smoothing results as well as baseline algorithms that can generate competitive edge-preserving smoothing results for a wide range of image contents. The established dataset contains 500 training and testing images with a number of representative visual object categories, while the baseline methods in our benchmark are built upon representative deep convolutional network architectures, on top of which we design novel loss functions well suited for edge-preserving image smoothing. The trained deep networks run faster than most state-of-the-art smoothing algorithms with leading smoothing results both qualitatively and quantitatively. The benchmark will be made publicly accessible.**

***Index Terms*—Edge-preserving smoothing, Benchmark, Image Dataset, Deep Convolutional Networks**

I. Introduction

In many image analysis and manipulation tasks, such as contour detection, image segmentation, and image stylization, it is important to preserve major image structures, such as salient edges and contours, while smoothing insignificant details. This can be achieved by edge-preserving image smoothing, a fundamental problem in image processing and low-level computer vision. Though a number of algorithms with diverse design philosophies have been proposed [1]–[12], there exist three problems that hinder the further development of edge-preserving image smoothing algorithms.

First, the performance evaluation of edge-preserving smoothing algorithms remains subjective. At present, the prevailing method is visual inspection by subjects on the smoothed images. Such an approach is time-consuming and cannot be applied in automatic systems. There lacks an objective metric to evaluate the edge-preserving smoothing algorithms.

Second problem is that an edge-preserving smoothing algorithm is typically evaluated on a very small image set against other algorithms. There lacks a widely accepted large-scale image database for algorithm evaluation. While a smoothing algorithm produces impressive results on certain types of images, it may not perform well on other types of images. Thus, a large database for a holistic evaluation of edge-preserving smoothing algorithms is much needed.

Third, smoothing algorithms typically have tunable parameters and images with different categories of contents need different parameter settings. To the best of our knowledge, no smoothing algorithms can perform reasonably well on a wide range of image contents using a single parameter setting.

To address the aforementioned problems, in this paper we propose a benchmark for edge-preserving image smoothing. This benchmark includes an image dataset with "groundtruth" image smoothing results as well as baseline models that are capable of generating reasonable edge-preserving smoothing results for a wide range of image contents. Our image dataset contains 500 training and testing images with a number of visual object categories, including humans, animals, plants, indoor scenes, landscapes and vehicles. The groundtruth smoothing results in our dataset are not directly generated by handcraft approaches, but manually chosen from results generated by existing state-of-the-art edge-preserving smoothing algorithms. This is justified by two reasons. First, as discussed earlier, a single state-of-the-art smoothing algorithm is capable of producing high-quality smoothing results over a small range of image contents especially when its parameters have been fine-tuned. Therefore, a collection of smoothing algorithms are able to generate high-quality results over a wide range of contents. The only caveat is that the best results generated by these algorithms for a specific image need to be hand-picked by humans. Second, since an image has hundreds of thousands of pixels, directly annotating pixelwise smoothing results by humans is too labor-intensive and error-prone.

To establish the baseline algorithms in our benchmark, we resort to the latest deep neural networks. Deep neural networks have a large number of parameters (weights). Once these weights have been trained, they can be fixed and the resulting network has very strong generalization capability and can deal with different types of inputs. Thus, a trained deep neural network on edge-preserving smoothing dataset is expected to

F. Zhu is with the Department of Computer Science, The University of Hong Kong, Hong Kong. e-mail: zhufeida@connect.hku.hk.

Z. Liang and L. Zhang are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

X. Jia is with School of Mathematics and Statistics, Xidian University, Xi'an, China.

Y. Yu is with the Department of Computer Science, The University of Hong Kong and Deepwise AI Lab.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes one PDF file of image results. Contact zhufeida@connect.hku.hk for further questions about this work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

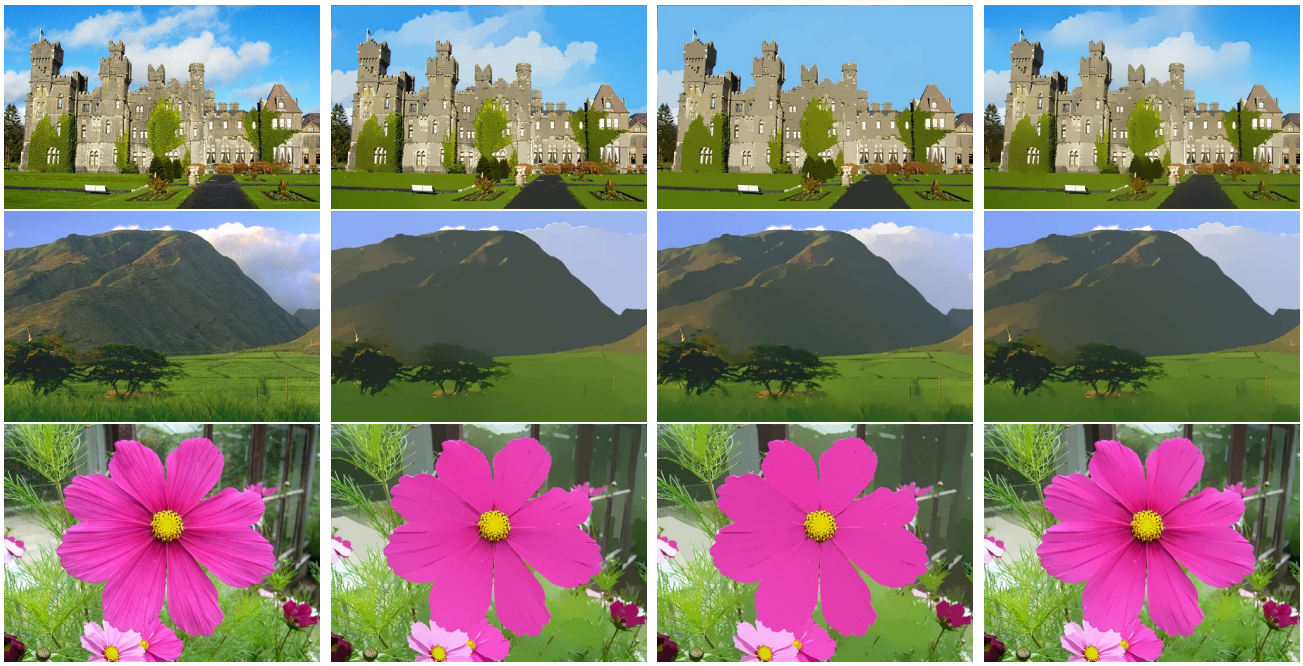


Fig. 1. In our dataset, each source image is associated with 14 edge-preserving smoothing results selected by different subjects from seven edge-preserving smoothing algorithms. The first column shows three source images. On the right three columns, we present 3 human-selected results for each source image.

perform consistently well in spite of the diverse image contents, which is the goal we want to achieve for edge-preserving image smoothing. We also note that deep learning has been broadly applied to low-level computer vision problems and has achieved state-of-the-art results. Examples include reproducing edge-preserving filters [13]–[15], image denoising [16], [17], image super-resolution [18]–[22], and JPEG deblocking [17], [23]. Specifically, we use the following two existing representative network architectures as our baseline methods, very deep convolutional networks (VDCNN) and deep residual networks (ResNet). On top of these network architectures, we design novel loss functions well suited for edge-preserving image smoothing. The deep networks trained over our dataset run faster than most state-of-the-art edge-preserving smoothing algorithms, while the smoothing performance of our ResNet-based model outperforms these algorithms both qualitatively and quantitatively. Our benchmark will be publicly released.

The remainder of this paper is organized as follows. Section 2 reviews the prior work in the research. Section 3 describes the construction of our dataset in detail. The objective metric for edge-preserving smoothing is presented in Section 4. Section 5 describes our baseline deep learning model for the smoothing task. In Section 6, we verify our benchmark by applying our baseline model to the tone mapping and contrast enhancement tasks. Section 7 is the conclusion.

II. Related Work

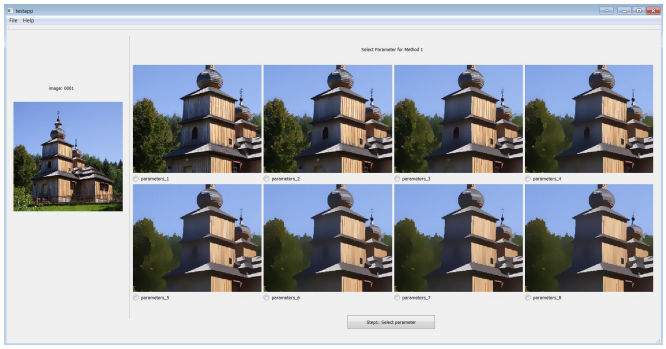
Edge-Preserving Smoothing: Many methods have been proposed for on edge-preserving smoothing, which can be categorized into two groups. The first group is local filter based approaches, where the filters are designed based on image statistics within a local window. Representative filters include Bilateral Filter [1], Weighted Median Filter (WMF) [10], Anisotropic Diffusion (AD) [2], and Edge-avoiding Wavelet (EAW) [4]. Rolling Guidance Filter (RGF) [5] applies the weighted filters iteratively, and Tree Filtering [9] utilizes minimum spanning tree to smooth out details while preserving major structure. However, the local filters have a common limitation in that they often introduce artifacts (such as halos along the edge) because only the local image statistics are used in the filtering, and we cannot explicitly control the statistical properties of the filtered images.

The second group is global optimization based approaches. The smoothed image is obtained by solving a global objective function, which usually involves a data term, constraining the distance between original image and smoothed image, and a regularization term, striving to achieve smoothness. Representative methods include Weighted Least Square smoothing (WLS) [3], L_0 smoothing [7], Fast Global Smoother (FGS) [8], L_1 smoothing [12] and SD filter [6]. Such methods overcome several limitations of local filter based approaches such as halos and gradient reversals. However, increased computational cost comes with solving the large-scale linear systems [3], [12].

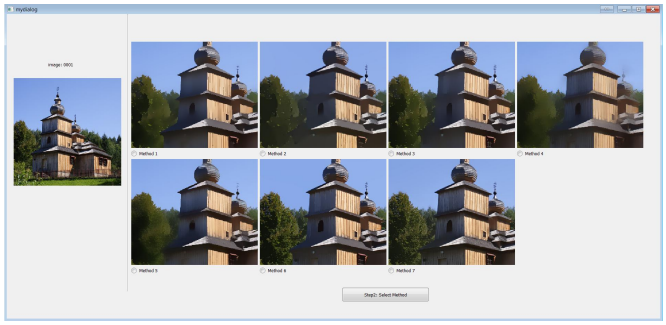
Quantitative Evaluation: Bao *et al*. [9] applied different edge-preserving filters to the test images in Berkeley Segmentation Dataset (BSDS300) [24] prior to boundary detection. F-measure is used to evaluate the filters' effectiveness in suppressing trivial details and preserving edges. Ham *et al*. [6], [25] proposed ODS and OIS [26] using the gradient magnitudes of filtered images to measure the effectiveness of filters. However, these evaluation approaches may suffer from the deviation of detected boundaries. In contrast, we construct a dataset of source images and their associated "groundtruth" (human-selected edge-preserving smoothed images). Objective evaluation can be performed by directly comparing the results

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

Fig. 2. Sample source images from our dataset for edge-preserving image smoothing.



(a) Step 1: Choose the best result for each algorithm



(b) Step 2: Choose the best result from 7 algorithms

Fig. 3. Snapshots of our two-step selection interface.

of an edge-preserving filter against such "groundtruth".

Deep Edge-aware Filter Learning: Xu *et al*. [13] proposed a method to learn and reproduce individual edge-preserving filters. They used a convolutional neural network to predict smoothed image gradients and then run an expensive step to reconstruct the smoothed image itself. However, the β parameter in their reconstruction step is not fixed and varies with different filters. Liu *et al*. [14] proposed a hybrid network by incorporating spatially varying recurrent neural networks (RNN) conditioned on the input image. A deep CNN is used to learn the weight map of the RNN. Li *et al*. [15] proposed a learning-based method to construct a CNN-based joint filter to transfer the structure of a guidance image to a target image for structure-texture separation. Fan *et al*. [27] exploited edge information by separating the image smoothing problem into two steps. The first sub-network is supervised to predict the edge map and the second sub-network reconstructs the target image by leveraging the predicted edge map.

In contrast, we do not aim to reproduce individual filters but the best results among a number of filters over a wide range of image contents. Furthermore, our deep neural networks learn pixelwise colors in the smoothed result instead of smoothed image gradients. Thus an input image can be smoothed efficiently with a single forward pass through the network without the need of a gradient-based reconstruction step.

III. A Dataset for Edge-Preserving Smoothing

A major source of our images is the database reported in [28]. This database is composed of a large number of high-quality natural images, which were originally employed for comparing image quality models. Another source of our dataset is the Berkeley Segmentation Dataset (BSDS500) [26] which has been widely used in the computer vision community. We manually chose 500 images of a variety of objects and scenes (humans, animals, plants, indoors, landscape, vehicles, etc.). Some sample images are shown in Figure 2. The selected images in the dataset contain clear structures and visible details which are well suited for evaluating edge preserving smoothing algorithms. Besides, the dataset are balanced among different objects and scenes. The images not suitable for edge-preserving smoothing (e.g., images without clear structures but filled with textured regions) are excluded from our dataset.

As mentioned earlier, the groundtruth smoothing results in our dataset are not directly annotated by humans, but manually chosen from results generated by existing state-of-the-art edge-preserving smoothing algorithms. This is because an image has hundreds of thousands of pixels, and thus directly annotating pixelwise smoothing results by humans is too labor-intensive and error-prone. On the other hand, while a single state-of-the-art algorithm is capable of producing high-quality smoothing results over a certain range of image contents with fine-tuned parameters, a collection of different smoothing algorithms are able to generate high-quality results over a wide range of contents.

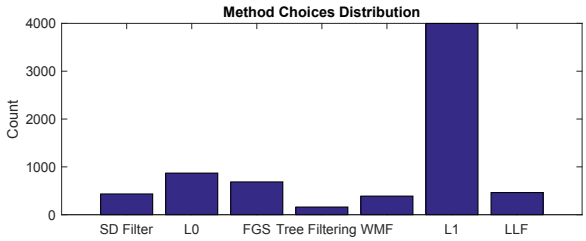
A. Selection Tool

We chose seven state-of-the-art and representative edge-preserving algorithms to construct our dataset, including SD filter [6], L_0 smoothing [7], Fast Global Smoother (FGS) [8], Tree Filtering [9], Weighted Median Filter (WMF) [10], L_1 smoothing [12] and Local Laplacian filter (LLF) [11]. The selection considerations of these filters are twofold. The first is the representativeness and impact of the work (e.g., high citations). This consideration ensures that the selected filters are state-of-the-arts. The second consideration is the diversity to leverage the merits of different types of filters. Based on these considerations, we selected 4 global methods [6]–[8], [12] and 3 local methods [9]–[11]. The global filters explicitly formulate the edge-preserving smoothing process as a global optimization problem, while the local filters apply a weighted function depending on the similarity of features within a local window.

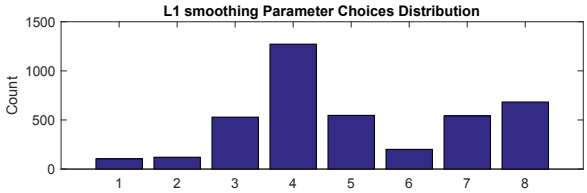
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

TABLE I
WE PREDEFINED 8 SETS OF PARAMETERS FOR EACH SMOOTHING ALGORITHM. ADDITIONAL PARAMETERS ARE SET TO DEFAULT VALUES SUGGESTED BY THE ORIGINAL AUTHORS.

Parameters		1	2	3	4	5	6	7	8
SD filter	λ	1	5	15	30	50	70	90	110
L_0 smoothing	λ	0.005	0.01	0.02	0.03	0.04	0.05	0.06	0.08
FGS	σ_c	0.02	0.02	0.025	0.025	0.03	0.03	0.04	0.04
	λ	400	900	600	900	500	900	400	1200
Tree Filtering	σ	0.05	0.05	0.1	0.1	0.2	0.2	0.4	0.4
	σ_s	8	4	8	4	8	4	8	4
WMF	σ	10	30	50	70	90	110	130	150
L_1 smoothing	α	10	10	20	20	100	100	200	200
	θ	200	50	200	50	200	50	200	50
LLF	σ_r	0.1	0.1	0.2	0.2	0.4	0.4	0.6	0.6
	α	2	4	2	4	2	4	2	4



(a) Vote distribution among different algorithms



(b) Vote distribution among different parameter settings of L_1 smoothing

Fig. 4. Distributions of human selection results. Two distributions of users' votes over different smoothing algorithms and over different parameter settings of the algorithms of L_1 smoothing.

According to the fact that the best smoothing results of different images may come from different algorithms with different parameter settings, we have developed an interactive interface where one can choose the proper edge-preserving smoothing result in two steps:

• Step 1: Given a source image, choose a parameter setting for each algorithm which generates the best smoothing result for that algorithm .

• Step 2: Choose the best one from the best results of the seven algorithms.

Figure 3 shows snapshots of the selection tool. The source image is shown in the top left corner. We pre-defined eight parameter settings for each algorithm, as shown in Table I. These eight settings are selected to cover a wide range of coarse-to-fine smoothing levels. For each algorithm, the smoothing results corresponding to these eight parameter settings are presented on the same screen (Figure 3(a)). A user first chooses the best result produced by different parameters for each method. Afterwards, the seven smoothing results chosen from the previous step, one for each of the seven algorithms, are shown side by side on a pop-out window (Figure 3(b)). The user then chooses the best result from the seven as the final selection for the current source image.

Since the above two steps heavily rely on visual comparisons and multiple images need to be shown on the same screen, we use a 32-inch Truecolor IPS monitor with a high resolution (3840×2160) for dataset construction.

B. Selection Protocol

As is widely acknowledged, the general criterion for basic edge-preserving smoothing is that the visually salient edges of major structures should be preserved while the trivial details should be removed. Subjects are instructed to select good edge-preserving smoothing results from different edge-preserving smoothing (EPS) filters with different parameter settings.

Since humans could have different perceptual comprehension of trivial details and major structures, they might select different smoothed images as their preferred edge-preserving smoothing results. To model such perceptual differences among human users, we formed a subject group of 26 volunteers, all of whom are graduate students in The University of Hong Kong and The Hong Kong Polytechnic University. Each source image in our dataset and its associated smoothing results were randomly assigned to 14 volunteers. That is, on average one volunteer was asked to select proper smoothing images for 267 source images. Most of the volunteers do not have prior experiences on edge-preserving image smoothing. Simple but key instructions were given to the volunteers in order to familiarize them with the nature of edge-preserving smoothing.

• Instruction 1: Strong edges should be preserved and blurry effects at significant edges are extremely undesired.

• Instruction 2: The color of a smoothed image should be as close to the original image as possible.

• Instruction 3: Under instructions 1 and 2, the smoother, the better.

The volunteers were further presented with a few unambiguous examples of edge-preserving smoothing for training. It typically takes a volunteer around 2 minutes to select one final result. In order to minimize the negative effects of visual fatigue, only a single work session up to 60 minutes is allowed in any single day.

In addition, many types of EPS algorithms have been proposed or tailored to meet specific criterion of their targeted applications (e.g., tone mapping). In our proposed benchmark construction, we carefully selected 7 representative EPS algorithms, which are designed for various applications, to create the ground truths. The constructed dataset implicitly captures various good edge-preserving smoothing properties. As a consequence, the benchmark is applicable to a wide range of applications. We will demonstrate the effectiveness of our benchmark in applications of tone mapping and contrast enhancement in Section VI.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

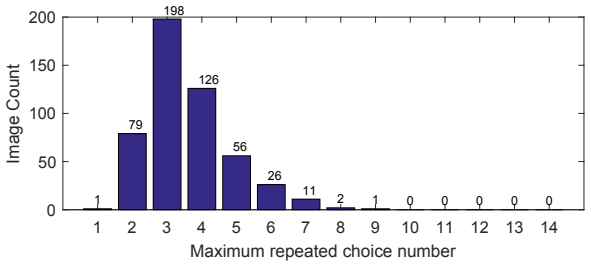


Fig. 5. Distribution of the maximum number of repeated choices.

TABLE II
THE MINIMUM WRMSE AND WMAE OF EXISTING STATE-OF-THE-ART EDGE-PRESERVING SMOOTHING METHODS AND DEEP MODELS. THE OPTIMAL PARAMETER SETTING OF EACH ALGORITHM IS USED ACROSS THE ENTIRE DATASET. RED, GREEN AND BLUE COLOR INDICATES THE BEST, SECOND BEST AND THIRD BEST RESULTS, RESPECTIVELY.

Error	SD filter	L_0 smooth	FGS	TreeFilter	WMF	L_1 smooth	LLF	VDCNN	ResNet
WRMSE*	11.57	10.64	10.67	14.31	11.83	9.89	11.06	9.78	9.03
WMAE*	7.65	6.93	6.82	9.24	7.96	5.76	7.29	6.15	5.55

C. Dataset Statistics

The constructed dataset for edge-preserving smoothing contains 500 natural images (400 for training and 100 for testing). As mentioned above, to reduce the bias of subject preference during manual selection, we collected 14 human-selected smoothing results for each image. The entire process lasted for one and half months.

Since each image is finally associated with 14 human-selected smoothing results, there are 7000 choices in total. As shown in Figure 4(a), a large proportion of the choices (3999 choices) are generated by the L_1 smoothing algorithm. Nevertheless, there are still a considerable number of choices (3001) distributed among the other 6 smoothing algorithms. Figure 4(b) shows the distribution of the parameter choices for the L_1 smoothing method. This distribution confirms the observation that the proper smoothing result of different images may come from different algorithms with different parameter settings.

To show the consistency of choices of different volunteers on the same source image, we compute the maximum number of repeated choices for each image. Let $count_t(m,p)$ denote the number of volunteers who chose method m with parameter setting p to compute the proper smoothing result of the t-th source image. Note that $\sum_{m=1}^{7}\sum_{p=1}^{8} count_t(m,p) = 14$. The maximum number of repeated choices for image t is defined as $\max_m \max_p count_t(m,p)$. The distribution of the maximum number of repeated choices across all images is shown in Figure 5. We can see that there are 420 (out of 500) images whose maximum number of repeated choices is greater than or equal to 3. In other words, for 84% of the source images, at least 3 volunteers chose the result from the same algorithm with the same parameter.

IV. QUANTITATIVE MEASURES

Denote by $x^t_{i,j}$ the pixel value at position (i,j) of the t-th source image in our dataset and denote by $y^{t,k}_{i,j}$ the pixel value of the corresponding "groundtruth" smoothed image selected by the k-th volunteer ($k \in [1,2,...,14]$) . We measure the quality of an edge-preserving filter F in terms of the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). In our problem, there are multiple "groundtruth" smoothed images selected by different subjects, and we define the RMSE and MAE as follows:

$$RMSE = \left(\frac{\sum_t \sum_{i,j} \sum_{k=1}^{14} \frac{1}{14} ||F(x^t)_{i,j} - y^{t,k}_{i,j}||^2}{\sum_t \sum_{i,j}} \right)^{\frac{1}{2}} \quad (1)$$

$$MAE = \frac{\sum_t \sum_{i,j} \sum_{k=1}^{14} \frac{1}{14} ||F(x^t)_{i,j} - y^{t,k}_{i,j}||_1}{\sum_t \sum_{i,j}} \quad (2)$$

where $F(x^t)_{i,j}$ is the pixel value in the smoothed image produced by the edge-preserving filter F. The denominator $\sum_t \sum_{i,j}$ denotes the total number of pixels in all images.

Due to the subjective nature of image quality assessment, inevitably there exist noises and outliers in the "groundtruth" smoothed images selected by different subjects. To reduce the effect of noises and outliers on performance evaluation, we take a voting strategy to focus on those smoothed images chosen by more subjects.

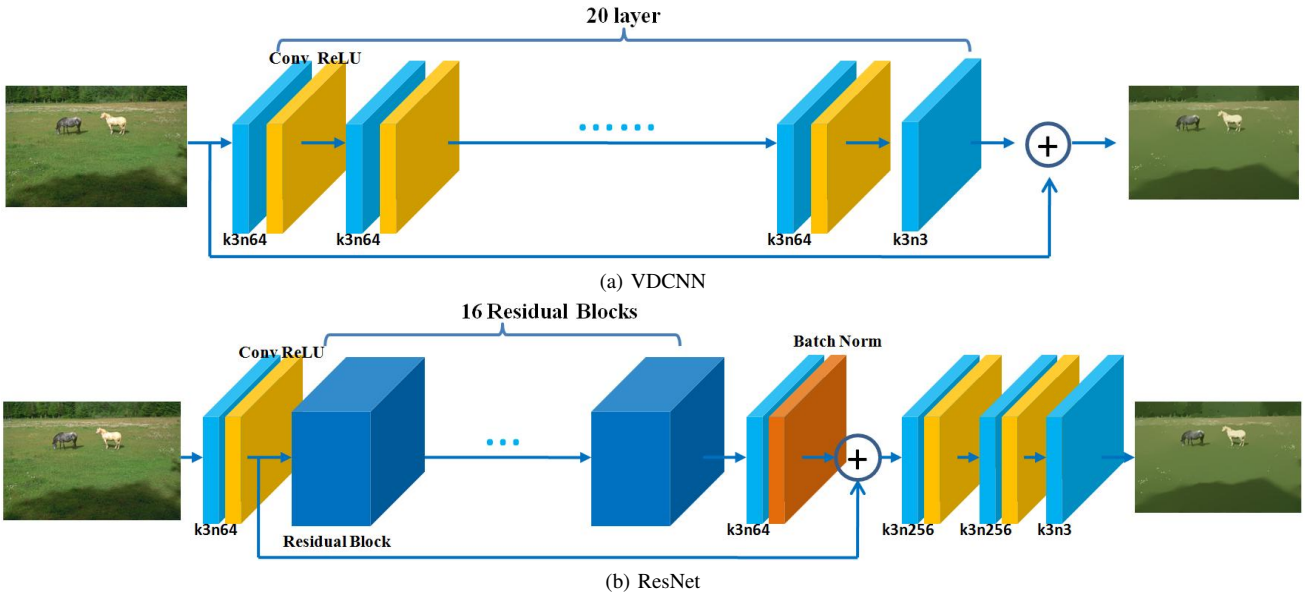
As mentioned in Section III-C, $count_t(m,p)$ denotes the number of subjects who chose method m with parameter setting p as the best smoothing filter of the t-th source image. Each image is associated with 14 human-selected smoothing results, and there are 7000 choices in total. The total number of times that method m with parameter setting p was chosen by a subject is denoted by COUNT(m,p).

The voting strategy is that for each source image, we sort its choice numbers ($count_t(m,p)$) in a descending order. If there is a tie between different combinations of methods and parameter settings, we sort them according to COUNT(m,p). That is because the total number of times a method with one of its parameter settings was chosen is an indicator of its overall performance. We tend to choose the smoothed images produced by more reliable methods when user preferences are the same. We only keep the first five results "groundtruth" smoothed images. For example, if method m with parameter p was selected by most subjects for the t-th source image, we let $Y^{t,1}$ denote the smoothed image produced by method m and parameter p, and set $count(Y^{t,1}) = count_t(m,p)$. We denote by $Y^{t,2}$ the second most frequently chosen smoothed image, and so on. The quantitative measures defined in Equations 1-2 can be extended to the weighted RMSE (WRMSE) and weighted MAE (WMAE) as follows:

$$WRMSE = \left(\frac{\sum_t \sum_{i,j} \sum_{k=1}^{5} w_{t,k} ||F(x^t)_{i,j} - Y^{t,k}_{i,j}||^2}{\sum_t \sum_{i,j}} \right)^{\frac{1}{2}} \quad (3)$$

$$WMAE = \frac{\sum_t \sum_{i,j} \sum_{k=1}^{5} w_{t,k} ||F(x^t)_{i,j} - Y^{t,k}_{i,j}||_1}{\sum_t \sum_{i,j}} \quad (4)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing



(a) VDCNN

(b) ResNet

Fig. 6. Network architecture of (a) VDCNN and (b) ResNet. Each convolutional layer is denoted with kernel size (k) and number of feature maps (n). The stride is 1 for all convolutional layers. Residual block is illustrated in Figure 7.

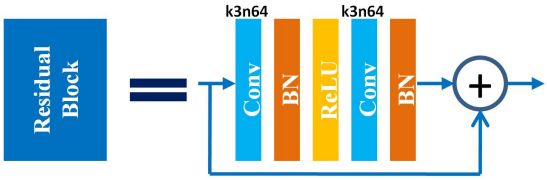


Fig. 7. Internal structure of a residual block used in ResNet.

where $w_{t,k}$ is defined as:

$$w_{t,k} = \frac{count(Y^{t,k})}{\sum_{k=1}^{5} count(Y^{t,k})} \tag{5}$$

In our quantitative measures, the weight of a combination of a method and a parameter setting varies across different source images because none of the existing smoothing algorithms performs consistently well over a wide range of image contents. The proposed quantitative measures assign higher weights to "groundtruth" smoothed images chosen by more subjects while excluding noises and outliers at the same time. We use Equations 3 and 4 to quantitatively measure the performance of various edge-preserving smoothing algorithms as well as our trained models in the rest of this paper.

A. Evaluation of Existing Algorithms

The parameter setting of a smoothing algorithm affects its performance. We measure WRMSE and WMAE across the entire testing set of our dataset with different parameter settings for each of the seven chosen algorithms. The minimum WRMSE and WMAE are denoted by WRMSE* and WMAE*, respectively, and the optimal parameter setting of each method was determined by greedy search. The process is as follows: set a group of parameter settings for each algorithm; apply them to all testing images; record WRMSE and WMAE. The parameter settings that make the seven chosen methods achieve their WRMSE* are given as follows: SD filter ($\lambda = 5$), L_0 smoothing (λ=0.02), FGS ($\sigma_c = 0.025, \lambda = 600$), Tree Filtering ($\sigma = 0.05, \sigma_s = 8$), WMF ($\sigma$=30), L_1 smoothing ($\alpha = 20, \theta = 50$), LLF ($\sigma_r = 0.4, \alpha = 2$). The parameter settings that make the seven methods achieve their WMAE* are given as follows: SD filter ($\lambda = 5$), L_0 smoothing (λ=0.01), FGS ($\sigma_c = 0.025, \lambda = 600$), Tree Filtering ($\sigma = 0.05, \sigma_s = 8$), WMF ($\sigma$=50), L_1 smoothing ($\alpha = 20, \theta = 50$), LLF ($\sigma_r = 0.2, \alpha = 4$). Additional parameters are set to default values suggested by original authors.

From Table II, we can see that the L_1 smoothing algorithm has lower WRMSE* and WMAE* than other smoothing algorithms because the results generated by the L_1 smoothing algorithm were most frequently chosen by the volunteers as their preferred results when our dataset was constructed, as shown in Figure 4.

V. Deep Learning Models

Deep neural networks have achieved great successes in low-level computer vision problems, including reproducing edge-preserving filters [13]–[15], image denoising [16], [17], image super-resolution [18]–[22], and JPEG deblocking [17], [23]. To build the baseline models in our benchmark, we resort to the latest deep neural networks as learning-based baseline algorithms. Deep neural networks have a large number of parameters (weights), which can be optimized to address a specific task. A well-trained deep neural network on our dataset is expected to be able to produce high-quality results for a wide range of inputs. Thus, it is not necessary to tune parameters of the trained model for a new image, which is a desirable property for edge-preserving image smoothing.

In this section, we present two representative network architectures and report their performance as a baseline for our edge-preserving smoothing dataset. Specifically, we em-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

Fig. 8. Example of smoothing outputs using different losses. We can see that the deep model trained with $loss_{l_1} + loss_{nb}$ produces smoother result at sky, roof and grass regions than $loss_{l_2}$ or $loss_{l_1}$ alone.

ploy representative deep convolutional neural network (CNN) architectures for our problem. This is because recently, deep CNNs have been successfully used in many low-level vision tasks [16]–[18], [20]–[22] and their network architectures can be employed for different tasks, including edge-preserving smoothing.

A. Network Architecture

VDCNN: 20 layers are stacked to form a very deep convolutional neural network, as shown in Figure 6(a). The layers maintain the same spatial resolution, the same kernel size (3×3) and the same number of feature maps (64) except the last output layer which has 3 channels only. The original model proposed by Kim *et al*. [18] works on the luminance channel of an image, which is the common practice in the literature of single image super-resolution. In contrast, we perform the training on three RGB channels since all color channels change during edge-preserving smoothing.

ResNet: Residual networks [19], [29], [30] have exhibited outstanding performance in both low-level and high-level computer vision problems. As shown in Figure 7, a basic residual block includes convolutional layers (Conv), batch normalization (BN), rectified linear units (ReLU) and a skip connection. A complete ResNet architecture is shown in Figure 6(b). It was originally proposed in [19], where it is used for inferring photo-realistic high-resolution images from low-resolution ones. We replace ParametricReLU [31] by ReLU and remove the up-sampling layer here.

TABLE III
PERFORMANCE COMPARISON BETWEEN THE TWO BASELINE NETWORK ARCHITECTURES UNDER DIFFERENT LOSS FUNCTIONS.

Error	VDCNN			ResNet		
	$loss_{l_2}$	$loss_{l_1}$	$loss_{l_1} + loss_{nb}$	$loss_{l_2}$	$loss_{l_1}$	$loss_{l_1} + loss_{nb}$
WRMSE	10.14	9.90	9.78	9.58	9.51	9.03
WMAE	6.92	6.20	6.15	6.50	6.12	5.55

B. Loss Functions

Since there exist multiple "groundtruth" smoothed images for each source image in our dataset, we define a weighted L_2 loss (Equation 6) and a weighted L_1 loss (Equation 7) in a manner similar to the weighted RMSE in Equation 3 and the weighted MAE in Equation 4:

$$loss_{l_2} = \sum_t \sum_{i,j} \sum_{k=1}^{5} w_{t,k} \left\| M_\theta(x^t)_{i,j} - Y_{i,j}^{t,k} \right\|^2, \quad (6)$$

$$loss_{l_1} = \sum_t \sum_{i,j} \sum_{k=1}^{5} w_{t,k} \left\| M_\theta(x^t)_{i,j} - Y_{i,j}^{t,k} \right\|_1, \quad (7)$$

where $M_\theta(x^t)$ represents the output from a deep network and x^t is the input image.

In addition to the weighted L_2 loss and weighted L_1 loss which enforce the consistency between the predicted smoothed images and their corresponding groundtruth smoothed images, we propose a neighborhood loss (Equation 8) to explicitly encourage a network to learn the local variations of the groundtruth images:

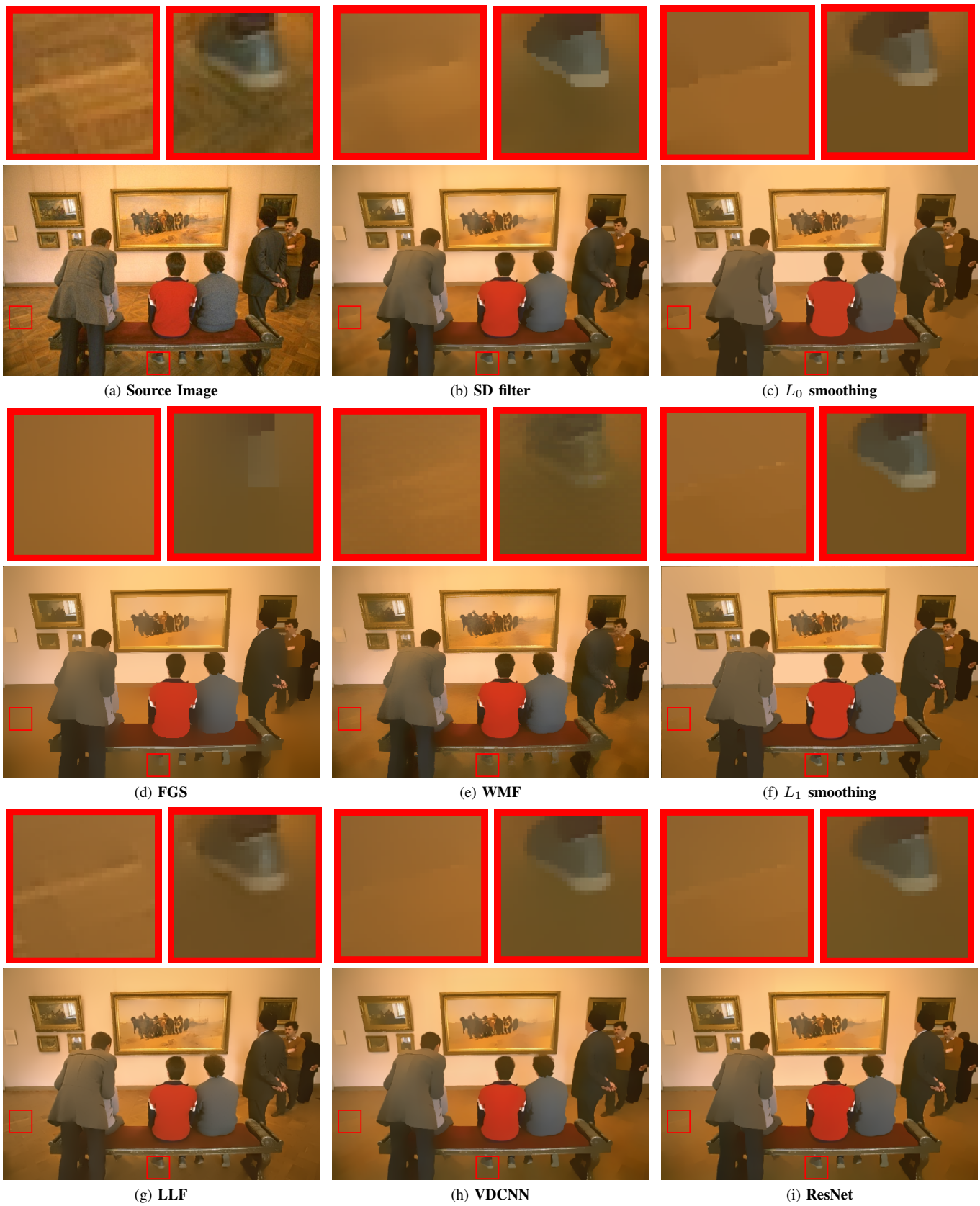
$$loss_{nb} = \sum_t \sum_{i,j} \sum_{k=1}^{5} \sum_{(p,q) \in N_{i,j}} w_{t,k} \left\| (M_\theta(x^t)_{i,j} - M_\theta(x^t)_{p,q}) - (Y_{i,j}^{t,k} - Y_{p,q}^{t,k}) \right\|_1, \quad (8)$$

where $N_{i,j}$ denotes the 5×5 neightborhood centered at pixel (i, j). The neighborhood term $||(M_\theta(x^t)_{i,j} - M_\theta(x^t)_{p,q}) - (Y_{i,j}^{t,k} - Y_{p,q}^{t,k})||_1$ explicitly penalizes deviations in the gradient domain. We add this neighborhood term to the weighted L_1 loss since edge-preserving smoothing involves evident gradient changes.

Quantitative results are presented in Table III, where we can see that ResNet achieves better performance than VDCNN when the same loss is used. $loss_{l_1} + loss_{nb}$ achieves better performance than $loss_{l_1}$ or $loss_{l_2}$ alone when the same network architecture is used, which validates the effectiveness of $loss_{nb}$. Figure 8 shows an example where we can see that the VDCNN trained with $loss_{l_1} + loss_{nb}$ produces smoother result at sky, roof and grass regions than $loss_{l_2}$ or $loss_{l_1}$ alone.

We thus take the VDCNN model and ResNet model trained with $loss_{l_1} + loss_{nb}$ as our baseline algorithms for our edge-preserving smoothing benchmark.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing



(a) **Source Image** (b) **SD filter** (c) L_0 **smoothing**

(d) **FGS** (e) **WMF** (f) L_1 **smoothing**

(g) **LLF** (h) **VDCNN** (i) **ResNet**

Fig. 9. Comparison of edge-preserving smoothing results by existing state-of-the-art algorithms and deep models. (a) Source Image. (b-g) Results by SD filter, L_0 smoothing, FGS, WMF, L_1 smoothing and LLF, respectively. The parameters are set as the optimal parameters for WMAE* illustrated in Section IV-A. (h) VDCNN with $loss_{l_1} + loss_{nb}$. (i) ResNet with $loss_{l_1} + loss_{nb}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

Fig. 10. Comparison between L_0 smoothing algorithm and our ResNet model. L_0 smoothing algorithm needs different parameter settings for the 'Racing car' and the 'Gloves' images. If we set λ=0.03 for the 'Racing car' image, the edge between grass and road will blur. λ=0.01 is the proper setting. However, if we set λ=0.01 for the 'Gloves' image, there will be undesirable noises. In contrast, our ResNet model produces robust visual results on different images without tuning parameters.

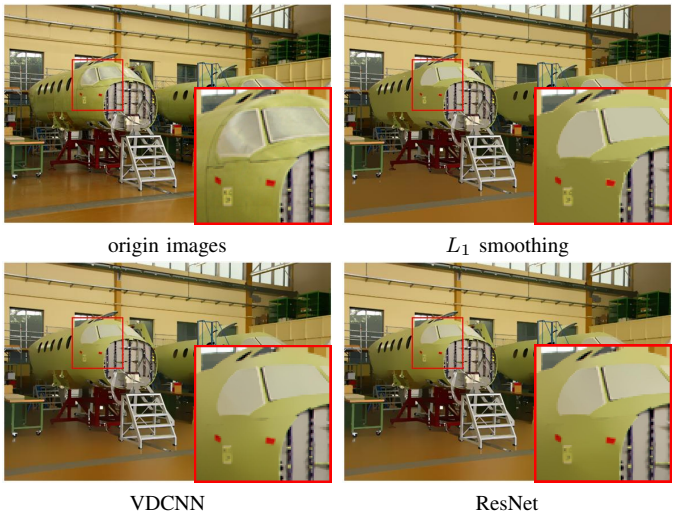


Fig. 11. Comparison between L_1 smoothing algorithm and deep models.

C. Network Training

We augment the training data with horizontal flips. RGB training patches are randomly sampled from source images and the corresponding smoothed images. We set different training patch sizes and mini-batch sizes for VDCNN and ResNet since they have different receptive fields and model complexity. For VDCNN, patch size is set to 41×41, and mini-batch size is set to 64. For ResNet, patch size is set to 96×96, and mini-batch size is set to 16.

Our deep models are trained using the ADAM optimizer [32] with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$. The initial learning rate is set to 10^{-3}. After the initial model converges, the learning rate is decreased by a factor of 10. Training is terminated once the model converges again.

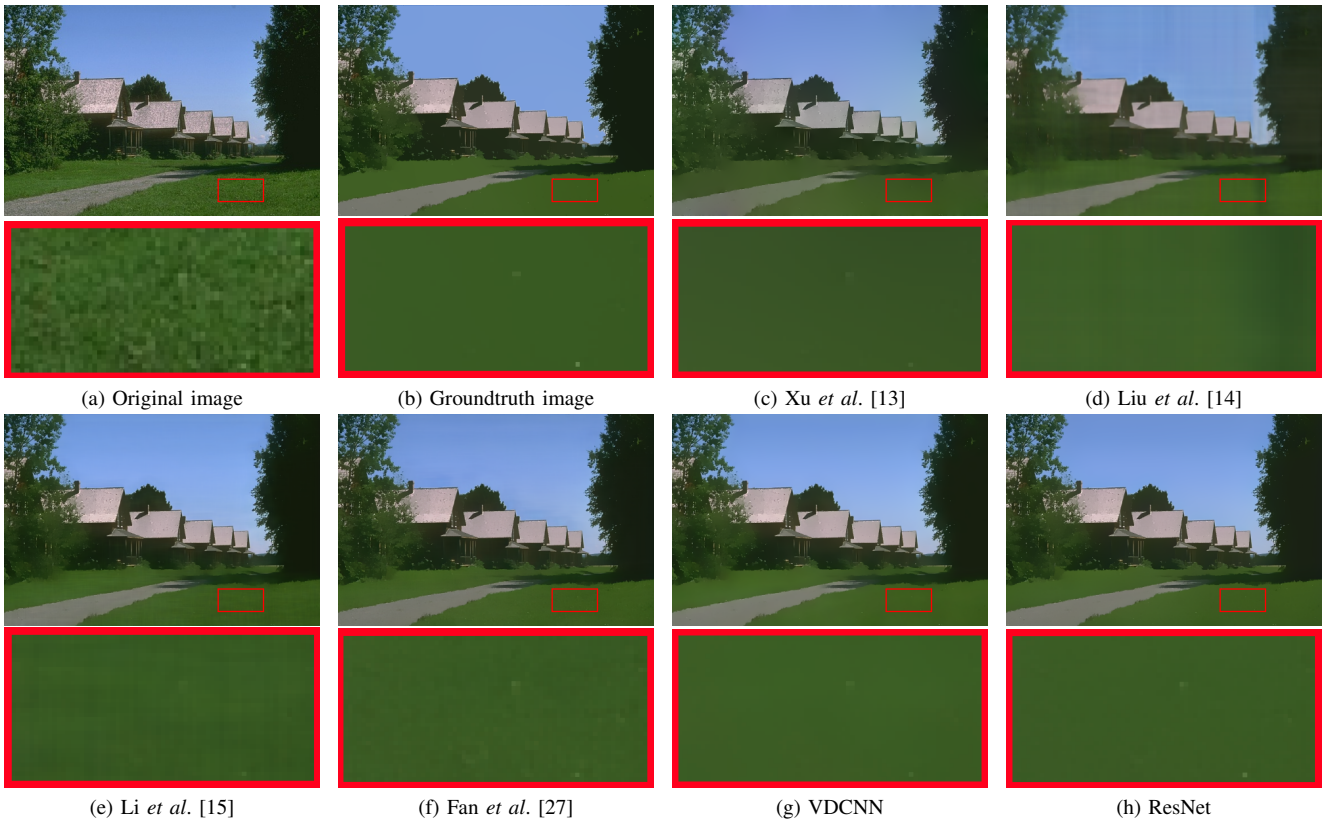
We implemented the VDCNN and ResNet models in the Tensorflow framework and trained them using NVIDIA GeForce GTX 1080TI GPU. It takes one day to train VDCNN and two days to train ResNet, respectively.

D. Evaluation

The VDCNN model and ResNet model trained with $loss_{l_1} + loss_{nb}$ are taken as the baseline algorithms for our edge-preserving smoothing benchmark. In this section, we report their performance both quantitatively and qualitatively.

As shown in Table II, the ResNet model achieves the lowest WRMSE and WMAE when compared to existing smoothing algorithms. The VDCNN model also achieves favorable performance, with slightly bigger errors than the ResNet model. An example is presented in Figure 9, where our two baseline models are compared with the existing edge-preserving smoothing algorithms. Please note that these smoothing algorithms use their optimal parameter settings for achieving WMAE*. It can be seen that some algorithms, such

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing



(a) Original image (b) Groundtruth image (c) Xu *et al*. [13] (d) Liu *et al*. [14]

(e) Li *et al*. [15] (f) Fan *et al*. [27] (g) VDCNN (h) ResNet

Fig. 12. Qualitative comparison with recent CNN-based methods [13]–[15], [27]. The groundtruth image shown in Figure (b) is the most frequently chosen image. It can be seen that [13] produces smoothed regions but suffers from color transitions. The method of [14] generates obvious artifacts at 'sky' and 'grass' regions. Methods in [15] and [27] produce generally visually pleasing results but there still exist unwanted details. Our ResNet result is visually closer to the groundtruth. **Best viewed with zoom on screen.**

as SD filter, L_0 smoothing, Tree Filtering and LLF, cannot effectively remove trivial details at the floor regions. Some algorithms, such as FGS and WMF, blur the area around the foot regions with salient edges. L_1 smoothing and the deep models achieve overall better quality of edge-preserving smoothing than other algorithms. More results can be found in the supplemental materials.

A more detailed visual comparison between the deep models and the L_1 smoothing algorithm [12] is given in Figure 11 considering the fact that the L_1 smoothing algorithm is the most frequently chosen algorithm and it achieves the lowest WRMSE and WMAE among existing state-of-the-art algorithms. From Figure 11 we can see that the L_1 smoothing algorithm wrongly increases the color contrast between two flattened regions on the airplane. The results from VDCNN and ResNet models do not have such artifacts.

As mentioned earlier, we do not aim to reproduce individual filters like [13]–[15], [27]. By utilizing the constructed dataset, our baseline algorithm aims to train a deep CNN model that can produce reasonable edge-preserving smoothing results for a wide range of image contents without further tuning parameters. To the best of our knowledge, existing smoothing algorithms cannot perform consistently well on a wide range of image contents using a single parameter setting. As an example, a comparison between L_0 smoothing [7] and our ResNet model is shown in Figure 10. We can see that the L_0 smoothing algorithm needs to set different parameters for the 'Racing car' and the 'Gloves' images. If we set λ=0.03 for the 'Racing Car' image, the edge between grass and road will blur. λ=0.01 is the proper setting for the 'Racing Car' image. However, if we set λ=0.01 for the 'Gloves' image, there still remain undesirable noises. In contrast, our ResNet model produces robust visual results on different images without tuning parameters. More results can be found in the supplementary file.

TABLE IV
RUN TIME (SECOND) OF EXISTING STATE-OF-THE-ART EDGE-PRESERVING SMOOTHING ALGORITHMS AND OUR DEEP MODELS.

Method	SD filter	L_0 smoothing	FGS	Tree Filtering	WMF
Run time	10.46	1.24	0.05	0.18	0.52
Method	L_1 smoothing	LLF	Our VDCNN	Our ResNet	
Run time	328	199	0.41	0.78	

E. Run Time

In addition to visual quality, testing speed is also an important aspect for image smoothing methods. We report the running time of existing smoothing algorithms using the author-provided Matlab code on a 3.4GHz Intel i7 processor. The average running time over 100 testing images is shown in Table IV. The L_1 smoothing algorithm has lower WRMSE* and WMAE* than other smoothing algorithms, but spends hundreds of seconds on solving a series of large-scale sparse

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

TABLE V
QUANTITATIVE COMPARISON WITH RECENT CNN-BASED METHODS INCLUDING XU *et al.* [13], LIU *et al.* [14], LI *et al*. [15] AND FAN *et al.* [27]. RUN TIME (SECOND) OF PREVIOUS METHODS AND OUR DEEP MODELS ARE ALSO REPORTED.

	[13]	[14]	[15]	[27]	Our VDCNN	Our ResNet
WRMSE	12.49	10.8	10.18	9.12	9.78	9.03
WMAE	9.27	7.3	6.57	5.74	6.15	5.55
Run time	1.4	0.24	0.35	0.62	0.41	0.78

linear systems. The slow processing speed of L_1 smoothing algorithm prevents it from being an ideal pre-processing tool for other image processing applications, e.g., edge detection. In contrast, our ResNet-based model achieves the lowest WRMSE and WMAE while its GPU implementation runs faster than most existing state-of-the-art smoothing algorithms.

F. Comparison with other CNN-based methods

Recently, CNN-based approaches [13]–[15], [27] have been proposed to reproduce individual smoothing filters. We also compare our method with those approaches on our proposed dataset. Note that [13]–[15], [27] are originally designed to mimic smoothing filters where the groundtruth smoothed images are produced by the target filter. However, each source image in our dataset is associated with five groundtruth smoothed images and the quantitative measures are defined as weighted RMSE and weighted MAE (Equations 3-4). For fair comparison, we modify the loss function of previous CNN-based methods to weighted loss function like what we define in Equations 6-7.

Table V presents the quantitative results of our methods and previous CNN-based methods on the 100 test images. It can be seen that our ResNet-based model achieves the lowest WRMSE and WMAE thanks to the deeper structure and novel neighborhood loss function. There are totally 37 convolutional layers in our ResNet model. In comparison, Xu *et al.* [13] proposed a 3-layer convolutional neural networks to learn the gradient map. Liu *et al.* incorporated convolutional neural networks in U-net style and recurrent neural networks together while the deep CNN consists of 9 layers. Li *et al.* [15] proposed a joint network architecture of three components. Each component is a three-layer network. Fan *et al*. [27] presented a Cascaded Edge and Image Learning Network (CEILNet). Both E-CNN and I-CNN consist of 32 convolutional layers and residual unit is implemented for the middle layers.

Figure 12 shows some visual results of previous CNN-based approaches. It can be seen that [13] produces smoothed regions but it suffers from color transitions since this method requires a reconstruction step from gradient domain to final image output. The method of [14] generates obvious artifacts at 'sky' and 'grass' regions. Methods in [15] and [27] produce generally visually pleasing results but there still exist unwanted details if we inspect closely. In contrast, our ResNet result is closer to the groundtruth.

TABLE VI
COMPARISON OF TMQI SCORES. TONE MAPPED IMAGE QUALITY INDEX (TMQI) [39] MEASURES THE STRUCTURAL FIDELITY AND STATISTICAL NATURALNESS.

	BF [33]	VAD [34]	LEP [35]	Ours
TMQI	0.9020	0.9041	0.8750	**0.9095**

TABLE VII
COMPARISON OF IEM. IMAGE ENHANCEMENT METRIC (IEM) [40] MEASURES THE IMPROVEMENT IN CONTRAST OF ENHANCED IMAGES.

	WV [36]	ND [37]	BF [1]	GF [38]	Ours
IEM	1.59	2.11	1.82	2.04	**2.18**

VI. APPLICATIONS

Smoothing high-contrast details while preserving edges is a useful step in many applications [33]–[35], [37], [38], [41]. In this section, we briefly discuss two applications, including tone mapping and contrast enhancement, by applying the trained ResNet model as the edge-preserving smoothing filter.

A. Tone mapping

Tone mapping is a popular technique to map one set of colors to another to reproduce the appearance of a high dynamic range (HDR) image on a low dynamic range (LDR) displayer. The state-of-the-art tone mappers commonly adopt a layer decomposition scheme to decompose the HDR image into low- and high-frequency layers and then process them separately. In particular, the low frequency layer is estimated by applying an edge-preserving filter to the original HDR image. The edge-preserving property is very important for avoiding halo artifact and achieving naturalness in the tone-mapped images.Thus, a stable and effective edge-preserving filter is highly desirable to improve the tone mapping performance.

To avoid halo artifact, an edge-preserving filter should be able to preserve the strong edge regions and flatten other regions in the image, regardless of the image contents and types. Our ResNet baseline model can handle this task well, because it is trained on our dataset which is constructed with such criteria. We use the tone mapping framework in [33] by replacing the original bilateral filter by our ResNet model. We compare the tone mapped results with several state-of-the-art tone mappers, including bilateral filter method (BF) [33], visual adaptation (VAD) [34], and local edge-preserving filter (LEP) [35]. BF-based tone mapper [33] may not be as effective as the recently proposed approaches, but BF is widely adopted in different image processing tasks. On the other hand, VAD [34] and LEP [35] are selected because they obtain state-of-the-art performance. We do not compare with [41] because saliency is beyond the scope of this work. Fig .13 shows our tone mapping results compared with these tone mappers. We can see that our tone mapper with ResNet model reaches an excellent balance between halo removal and naturalness preservation. Other tone mappers suffers from either halo artifact or over-enhancement problems.

To better investigate the performance of the competitive tone mappers, we collected 100 HDR images online for objective

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

Fig. 13. Comparison of tone mapping results. **From top to bottom: the HDR images, results by BF method [33], VAD method [34], LEP method [35] and our ResNet model.** We can see that the BF method may introduce halo artifacts at the border areas around the tree in the top image. The results produced by VAD method miss many details. The LEP method over-enhances the images and leads to unnatural results. In contrast, our results preserve the details and look natural. An objective evaluation is shown in Table VI.

evaluation. The TMQI metric [39] is used to score each tone mapped image by each method. Table VI shows the average TMQI score of each tone mapper. We can see that our tone mapper with ResNet model achieves the highest TMQI score. This demonstrates that our edge-preserving benchmark can facilitate the tone mapper to gain robust performance over different image types and contents.

B. Contrast enhancement

Contrast enhancement aims to enhance the local contrast of an image that suffers from large illumination variation. Similar to tone mapping, a proper contrast enhancement framework decomposes an image into two components, illumination and reflectance. The estimation of illumination requires a high-performance edge-preserving filter. In [37], the authors pro-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2908778, IEEE Transactions on Image Processing

Fig. 14. Comparison of contrast enhancement results for low-light images. **From top to bottom: the original images, results by WV method [36], ND method [37], BF method [1], GF method [38] and our ResNet model.**

posed a criterion for optimal contrast enhancement, based on which the edge-preserving filter should preserve the boundary regions of an image and flatten the texture regions as much as possible. This coincides with the criterion adopted in our benchmark.

To test our benchmark in the application of contrast enhancement, we adopt the algorithm framework in [37] and only replace the Diffusion-based filtering component by our ResNet model. We also compare with bilateral filter [1] and guided filter (GF) [38] since they have been widely used in different image processing tasks. The enhancement results together with the results of the state-of-the-art contrast enhancement methods including nonlinear diffusion method (ND) [37] and weighted variational method (WV) [36] are shown in Fig. 14. We can see that our enhancement results exhibit clearer structures and higher local contrast.

To measure the contrast enhancement results quantitatively, we report the Image Enhancement Metric (IEM) [40], a full reference IQA metric, to assess the contrast of the enhanced images. We used the 18 test images denoted as 'a' to 'r' in [42] for evaluation. Some results are shown in Fig. 14, and all results can be found in supplementary. As Table VII shows, the improved performance validates that the ResNet model trained on our benchmark can be adopted as an efficient edge-preserving smoothing filter for image contrast enhancement.

VII. Conclusions

We presented a benchmark for edge-preserving image smoothing for the purpose of quantitative performance evaluation and further advancing the state-of-the-art. This benchmark consists of 500 source images and their "groundtruth" image smoothing results as well as baseline learning models. The baseline models are representative deep convolutional network architectures, on top of which we design novel loss functions well suited for edge-preserving image smoothing. Our trained deep networks run fast at test time while their smoothing results outperform state-of-the-art smoothing algorithms both quantitatively and qualitatively.

Acknowledgments

This work was partially supported by Hong Kong Research Grants Council under General Research Funds (HKU17209714 and PolyU152124/15E).

The authors would like to thank Lida Li, Jin Xiao, Xindong Zhang, Hui Li, Jianrui Cai, Sijia Cai, Hui Zeng, Hongyi Zheng, Wangmeng Xiang, Shuai Li, Runjie Tan, Nana Fan, Kai Zhang, Shuhang Gu, Jun Xu, Lingxiao Yang, Anwang Peng, Wuyuan Xie, Wei Zhang, Weifeng Ge, Kan Wu, Haofeng Li, Chaowei Fang, Bingchen Gong, Sibei Yang and Xiangru Lin for constructing the dataset, and the reviewers for their insightful comments.

References

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[2] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[3] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 67.

[4] R. Fattal, "Edge-avoiding wavelets and their applications," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 22, 2009.

[5] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *European Conference on Computer Vision*. Springer, 2014, pp. 815–830.

[6] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[7] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l 0 gradient minimization," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6. ACM, 2011, p. 174.

[8] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast global image smoothing based on weighted least squares," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5638–5653, 2014.

[9] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang, "Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 555–569, 2014.

[10] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (wmf)," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2830–2837.

[11] S. Paris, S. W. Hasinoff, and J. Kautz, "Local laplacian filters: Edge-aware image processing with a laplacian pyramid." *ACM Trans. Graph.*, vol. 30, no. 4, pp. 68–1, 2011.

[12] S. Bi, X. Han, and Y. Yu, "An l 1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 78, 2015.

[13] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1669–1678.

[14] S. Liu, J. Pan, and M.-H. Yang, "Learning recursive filters for low-level vision via a hybrid neural network," in *European Conference on Computer Vision*. Springer, 2016, pp. 560–576.

[15] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," in *European Conference on Computer Vision*. Springer, 2016, pp. 154–169.

[16] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems*, 2016, pp. 2802–2810.

[17] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, 2017.

[18] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[19] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.

[21] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of International Conference on Computer Vision*, 2017.

[23] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 576–584.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[25] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4823–4831.

[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[27] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "A generic deep architecture for single image reflection removal and image smoothing," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[28] K. Ma, Q. Wu, Z. Wang, Z. Duanmu, H. Yong, H. Li, and L. Zhang, "Group mad competition-a new methodology to compare objective image quality models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1664–1673.

[29] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[33] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *ACM transactions on graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 257–266.

[34] S. Ferradans, M. Bertalmio, E. Provenzi, and V. Caselles, "An analysis of visual adaptation and contrast perception for tone mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2002–2012, Oct 2011.

[35] B. Gu, W. Li, M. Zhu, and M. Wang, "Local edge-preserving multiscale decomposition for high dynamic range image tone mapping," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 70–79, Jan 2013.

[36] X. Fu, D. Zeng, Y. Huang, X. P. Zhang, and X. Ding, "A weighted variational model for simultaneous reflectance and illumination estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2782–2790.

[37] Z. Liang, W. Liu, and R. Yao, "Contrast enhancement by nonlinear diffusion filtering," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 673–686, Feb 2016.

[38] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[39] H. Yeganeh and Z. Wang, "Objective quality assessment of tone-mapped images," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 657–667, Feb 2013.

[40] V. Jaya and R. Gopikakumari, "Iem: a new image enhancement metric for contrast and sharpness measurements," *International Journal of Computer Applications*, vol. 79, no. 9, 2013.

[41] Z. Li and J. Zheng, "Visual-salience-based tone mapping for high dynamic range images," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 7076–7082, 2014.

[42] H. Yue, J. Yang, X. Sun, F. Wu, and C. Hou, "Contrast enhancement based on intrinsic image decomposition," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3981–3994, 2017.

Feida Zhu received the B.Eng. degree in the Department of Automation from The University of Science and Technology of China, Hefei, China in 2014, and Ph.D degree in the Department of Computer Science, The University of Hong Kong, Hong Kong, in 2019. He is now working as a research fellow in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include image processing, computer vision and machine learning.

Zhetong Liang received the B.S. degree in electronic information science and technology from the Guangdong University of Technology, Guangzhou, China, in 2013, and the M.S. degree with the School of Electronic and Information Engineering, South China University of Technology, Guangdong, China. He is currently pursuing the PhD degree with the Department of Computing, School of Engineering, the Hong Kong Polytechnic University. His current research interests include computational imaging, image processing pipeline and deep learning.

Xixi Jia received the B.S., M.S. and PhD degrees from Xidian University, Xian, China, in 2012,2015 and 2018 respectively. He is now working at the School of Mathematics and Statistics, Xidian University. He was also worked as a research assistant at The Hong Kong Polytechnique University, HongKong from 2016-2017. His research interest covers image restoration, convex optimization and deep learning.

Lei Zhang (M'04, SM'14, F'18) received his B.Sc. degree in 1995 from Shenyang Institute of Aeronautical Engineering, Shenyang, P.R. China, and M.Sc. and Ph.D degrees in Control Theory and Engineering from Northwestern Polytechnical University, Xi'an, P.R. China, in 1998 and 2001, respectively. From 2001 to 2002, he was a research associate in the Department of Computing, The Hong Kong Polytechnic University. From January 2003 to January 2006 he worked as a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, McMaster University, Canada. In 2006, he joined the Department of Computing, The Hong Kong Polytechnic University, as an Assistant Professor. Since July 2017, he has been a Chair Professor in the same department. His research interests include Computer Vision, Image and Video Analysis, Pattern Recognition, and Biometrics, etc. Prof. Zhang has published more than 200 papers in those areas. As of 2019, his publications have been cited more than 39,000 times in literature. Prof. Zhang is a Senior Associate Editor of IEEE Trans. on Image Processing, and is/was an Associate Editor of SIAM Journal of Imaging Sciences, IEEE Trans. on CSVT, and Image and Vision Computing, etc. He is a "Clarivate Analytics Highly Cited Researcher" from 2015 to 2018. More information can be found in his homepage http://www4.comp.polyu.edu.hk/ cslzhang/.

Yizhou Yu (M'10, SM'12, F'19) received the PhD degree from University of California at Berkeley in 2000. He is a professor at The University of Hong Kong, and was a faculty member at University of Illinois at Urbana-Champaign for twelve years. He is a recipient of 2002 US National Science Foundation CAREER Award, 2007 NNSF China Overseas Distinguished Young Investigator Award, and ACCV 2018 Best Application Paper Award. Prof Yu has served on the editorial board of IET Computer Vision, The Visual Computer, and IEEE Transactions on Visualization and Computer Graphics. He has also served on the program committee of many leading international conferences, including SIGGRAPH, SIGGRAPH Asia, and International Conference on Computer Vision. His current research interests include computer vision, deep learning, biomedical data analysis, computational visual media and geometric computing.