Joint Image Filtering with Deep Convolutional Networks

Yijun Li, Jia-Bin Huang, *Member, IEEE*, Narendra Ahuja, and Ming-Hsuan Yang, *Senior Member, IEEE*

Abstract—Joint image filters leverage the guidance image as a prior and transfer the structural details from the guidance image to the target image for suppressing noise or enhancing spatial resolution. Existing methods either rely on various explicit filter constructions or hand-designed objective functions, thereby making it difficult to understand, improve, and accelerate these filters in a coherent framework. In this paper, we propose a learning-based approach for constructing joint filters based on Convolutional Neural Networks. In contrast to existing methods that consider only the guidance image, the proposed algorithm can selectively transfer salient structures that are consistent with both guidance and target images. We show that the model trained on a certain type of data, e.g., RGB and depth images, generalizes well to other modalities, e.g., flash/non-Flash and RGB/NIR images. We validate the effectiveness of the proposed joint filter through extensive experimental evaluations with state-of-the-art methods.

Index Terms—Joint filtering, deep convolutional neural networks, depth upsampling

1 Introduction

Image filtering with guidance signals, known as *joint* or *guided filtering*, has been successfully applied to numerous computer vision and computer graphics tasks, such as depth map enhancement [1], [2], [3], joint upsampling [1], [4], cross-modality noise reduction [5], [6], [7], and structure-texture separation [8], [9]. The wide applicability of joint filters can be attributed to the adaptability in handling visual signals in various image domains and modalities, as shown in Fig. 1. For a target image, the guidance image can either be the target image itself [6], [10], high-resolution RGB images [2], [3], [6], images from different sensing modalities [5], [11], [12], or filter outputs from previous iterations [9]. The basic idea behind joint image filtering is that we can transfer the important structural details contained in the guidance image to the target image. The main goal of joint filtering is to enhance the degraded target image due to noise or low spatial resolution while avoiding transferring extraneous structures that do not originally exist in the target image, e.g., texture-copying artifacts.

Several approaches have been developed to transfer structures in the guidance image to the target image. One category of algorithms is to construct joint filters for specific tasks. For example, the bilateral filtering algorithm [10] constructs spatially-varying filters that reflect local image structures (e.g., smooth regions, edges, textures) in the guidance image. Such filters can then be applied to the target image for edge-aware smoothing [10] or joint upsampling [4]. On the other hand, the guided image filter [6] assumes a locally linear model over the guidance image for filtering. However, these filters share one common drawback. That is, the filter construction considers only the information contained in the guidance image and remains fixed (i.e., static guidance). When the local structures in the guidance and target images are not consistent, these methods may transfer incorrect or extraneous contents to the target image.

To address this issue, recent efforts focus on considering the common structures existing in both the target and guidance images [7], [9], [13]. These frameworks typically build on iterative methods for minimizing global objective functions. The guidance signals are updated at each iteration (i.e., dynamic guidance) towards preserving the mutually consistent structures while suppressing contents that are not commonly shared in both images. However, these global optimization based methods often use hand-crafted objective functions that may not reflect natural image priors well and typically require a heavy computational load.

In this work, we propose a learning-based joint filter based on Convolutional Neural Networks (CNNs). We propose a network architecture that consists of three sub-networks and a skip connection, as shown in Fig. 2. The first two sub-networks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ extract informative features from both target and guidance images. These feature responses are then concatenated as inputs for the network $\mathrm{CNN_F}$ to selectively transfer common structures. As the target input and output images are largely similar, we introduce a skip connection, together with the output of $\mathrm{CNN_F}$ to reconstruct the filtered output. In other words, we enforce the network to focus on learning the residuals between the degraded target and the ground truth images. We train the network using large quantities of RGB/depth data and learn all the network parameters simultaneously without stage-wise training.

- *Y. Li and M.-H. Yang are with the School of Engineering, University of California, Merced, CA 95343. E-mail: {yli62, mhyang}@ucmerced.edu.*
- *J.-B. Huang is with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061. E-mail: jbhuang@vt.edu.*
- *N. Ahuja is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61820. E-mail: n-ahuja@illinois.edu.*

Manuscript received 3 Oct. 2017; revised 16 Dec. 2018; accepted 20 Dec. 2018. Date of publication 31 Dec. 2018; date of current version 11 July 2019. (Corresponding author: Ming-Hsuan Yang.) Recommended for acceptance by J. Jia. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2018.2890623

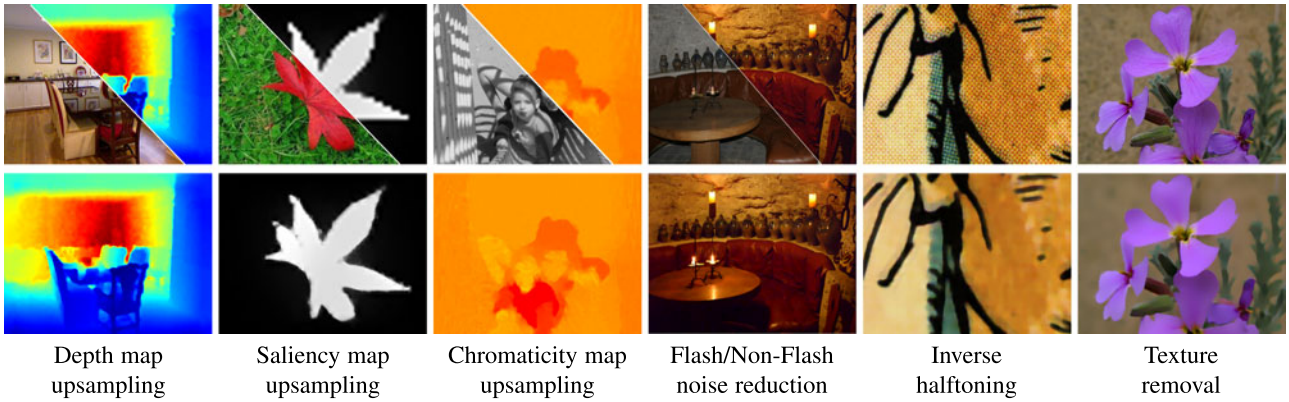


Fig. 1. *Sample applications of joint image filtering.* The target/guidance pair (top) can be various types of cross-modality visual data. With the help of the guidance image, important structures can be transferred to the degraded target image to help enhance the spatial resolution or suppress noises (bottom). The guidance image can either be high-resolution RGB images, images from different sensing modalities, or the target image itself.

Our algorithm differs from existing methods in that the proposed joint image filter is purely data-driven. This allows the network to handle complicated scenarios that may be difficult to capture through hand-crafted objective functions. While the network is trained using the RGB/depth data, the network learns how to selectively transfer structures by leveraging the prior from the guidance image, rather than predicting specific values. As a result, the learned network generalizes well for handling images in various domains and modalities.

We make the following contributions in this paper:

- We propose a learning-based framework for constructing a generic joint image filter. Our network takes both target and guidance images into consideration and naturally handles the inconsistent structure problem.
- Using the learned joint image filter for depth upsampling, we demonstrate the state-of-the-art performance on the NYU v2 [14] and SUN RGB-D [15] dataset and achieve competitive performance on the Middlebury dataset [16], [17].
- We show that the model trained on a certain type of data (e.g., RGB/depth) generalizes well to handle image data in a variety of domains.

A preliminary version of this work was presented earlier in [18]. In this paper, we significantly extend our work and summarize the main differences as follows. First, we propose an improved network architecture for joint image filtering. Instead of directly predicting filtered pixel values (as in [18]), we predict a residual image by adding a skip connection from the input target image to the output (Fig. 2). As the residual learning alleviates the need for restoring specific target image contents (which complicates the learning process), we show significant improvement in transferring accurate details from the guidance to the target image. Second, in [18], we train the model only using an RGB/depth dataset and then evaluate its generalization ability on other domains. In this work, we show that the model trained using an RGB/flow dataset also generalizes well on other visual domains. This demonstrates that our network design is insensitive to the modality of the training data. Third, we evaluate our approach on various joint image filter applications, compare against several state-of-the-art joint image filters (including concurrent work [19],

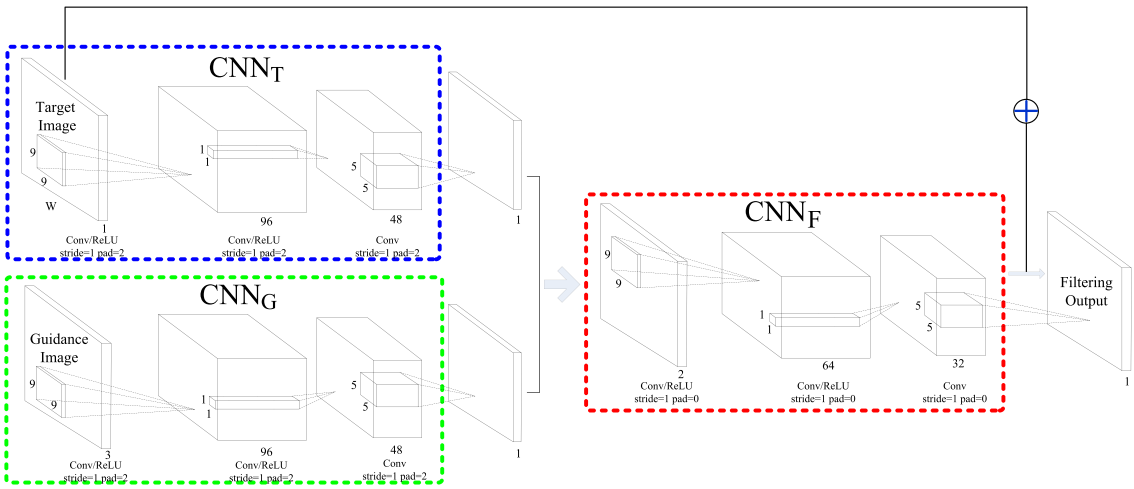


Fig. 2. *Network architecture for joint image filter.* The proposed deep joint image filter model consists of three major components. Each component is a three-layer network. The sub-networks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ aim to extract informative feature responses from the target and guidance images, respectively. We then concatenate these features responses together and use them as input for the network $\mathrm{CNN_F}$. In addition, we introduce a skip connection so that the network $\mathrm{CNN_F}$ learns to predict the residuals between the input target image and the desired ground truth output. We train the network to selectively transfer main structures while suppressing inconsistent structures using an RGB/depth dataset. While we describe these sub-networks individually, the parameters of all three sub-networks are updated simultaneously during the training stage.

[20]), and conduct a detailed ablation study by analyzing the performance of all methods under different hyper-parameter settings (e.g., filter number, filter size, network depth).

2 Related Work

Joint Image Filters. Joint image filters can be categorized into two main classes based on explicit filter construction or global optimization of data fidelity and regularization terms.

Explicit joint filters compute the filtered output as a weighted average of neighboring pixels in the target image. The bilateral filters [1], [4], [9], [10], [20], [21] and guided filters [6] are representative algorithms in this class. The filter weights, however, depend solely on the local structure of the guidance image. Therefore, erroneous or extraneous structures may be transferred to the target image due to the lack of consistency constraints. In contrast, our model considers the contents of both images based on feature maps and enforces consistency implicitly through learning from examples.

Numerous approaches formulate joint filtering based on a global optimization framework. The objective function typically consists of two terms: data fidelity and regularization terms. The data fidelity term ensures that the filtering output is close to the input target image. These techniques differ from each other mainly in the regularization term that encourages the output to have a similar structure with the guidance image. The regularization term can be defined according to texture derivatives [22], mid-level representations [2] such as segmentation and saliency, filtering outputs [13], or mutual structures shared by the target and guidance image [7]. However, global optimization based methods rely on hand-designed objective functions that may not reflect the complexities of natural images. Furthermore, these approaches involve iterative optimization are often time-consuming. In contrast, our method learns how to selectively transfer important details directly from the RGB/depth data. Although the training process is time-consuming, the learned model is efficient during run-time.

Learning-Based Image Filters. With significant success in high-level vision tasks [23], substantial efforts have been made to construct image filters using learning algorithms and CNNs. For example, the conventional bilateral filter can be improved by replacing the predefined filter weights with those learned from a large amount of data [24], [25], [26]. In the context of joint depth upsampling, Tai et al. [19] use a multi-scale guidance strategy to improve upsampling performance. Gu et al. [27] adjust the original guidance dynamically to account for the iterative updates of the filtering results. However, these methods [19], [27] are limited to the application of depth map upsampling. In contrast, our goal is to construct a generic joint filter for various applications using target/guidance image pairs in different visual domains.

Deep Models for Low-Level Vision. In addition to filtering, deep learning models have also been applied to other low-level vision and computational photography tasks. Examples include image denoising [28], raindrop removal [29], image super-resolution [30], image deblurring [31] and optical flow estimation [32]. Existing deep learning models for low-level vision use either one input image [28], [29], [30], [33] or two images in the same domain [32]. In contrast, our network can accommodate two streams of inputs by *heterogeneous* domains, e.g., RGB/NIR, flash/non-flash, RGD/Depth, intensity/color. Our network architecture bears some resemblance to that in Dosovitskiy et al. [32]. The main difference is that the merging layer used in [32] is a correlation operator while our model integrates the inputs through concatenating the feature responses. Furthermore, we adopt the residual learning by introducing the skip connection.

Another closely related work is by Xu et al. [33], which learns a CNN to approximate existing edge-aware filters from example images. Our method differs from [33] in two aspects. First, the goal of [33] is to use CNN for approximating existing edge-aware filters. In contrast, our goal is to learn a new joint image filter. Second, unlike the network in [33] that takes only one single RGB image, the proposed joint filter handles two images from different domains and modalities.

Skip Connections. As deeper networks have been developed for vision tasks, the information contained in the input or gradients can vanish and wash out by the time it reaches the end (or beginning) of the network. He et al. [34] address this problem through bypassing the signals from one layer to the next via skip connections. This residual learning method facilitates us to train very deep networks effectively. The work of [35] further strengthens its effectiveness with dense connections across all the layers. For low-level vision tasks, skip connection have been shown to be useful to restore high-frequency details [36], [37] by enforcing the network to learn the residual signals only.

3 Learning Joint Image Filters

In this section, we introduce a learning-based joint image filter based on CNNs. We first present the network design (Section 3.1) and skip connection (Section 3.2). Next, we describe the network training process (Section 3.3) and visualize the guidance map generated by the network (Section 3.4).

Our CNN model consists of three sub-networks: the target network $\mathrm{CNN_T}$, the guidance network $\mathrm{CNN_G}$, and the filter network $\mathrm{CNN_F}$ as shown in Fig. 2. First, the sub-network $\mathrm{CNN_T}$ takes the target image as input and extracts a feature map. Second, similar to $\mathrm{CNN_T}$, the sub-network $\mathrm{CNN_G}$ extracts a feature map from the guidance image. Third, the sub-network $\mathrm{CNN_F}$ takes the concatenated feature responses from the sub-networks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ as input and generates the residual, i.e., the difference between the degraded target image and ground truth. By adding the target input through the skip connection, we obtain the final joint filtering result. Here, the main roles of the two sub-networks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ are to serve as non-linear feature extractors that capture the local structural details in the respective target and guidance images. The sub-network $\mathrm{CNN_F}$ can be viewed as a non-linear regression function that maps the feature responses from both target and guidance images to the desired residuals. Note that the information from target and guidance images is simultaneously considered when predicting the final filtered result. Such a design allows us to selectively transfer structures and avoid texture-copying artifacts.

3.1 Network Architecture Design

To design a joint filter using CNNs, a straightforward implementation is to concatenate the target and guidance images

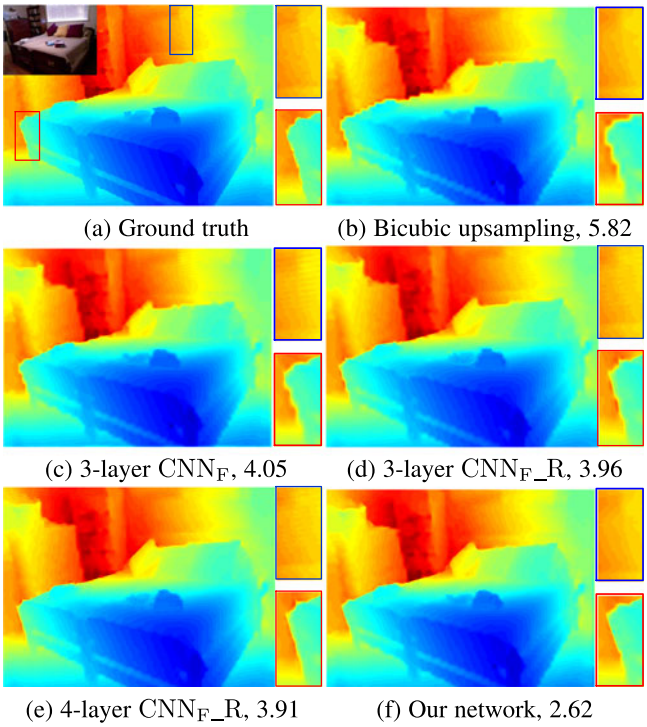


Fig. 3. *Comparison of network design.* Joint depth upsampling (8×) results of using different network architectures. (a) GT depth map (inset: Guidance image). (b) Bicubic upsampling. (c)-(e) Results from the straightforward implementation using $\mathrm{CNN_F}$ and $\mathrm{CNN_F}$_R. (f) Our results. Note the difference on the bed corner and curtain. The numbers are the RMSE metric based on the GT in (a).

together and directly train a generic CNN similar to the filter network $\mathrm{CNN_F}$. While in theory, we can train a generic CNN to approximate the desired function for joint filtering, our empirical results show that such a network generates poor results. Figs. 3c and 3d shows an example of joint depth upsampling using the network $\mathrm{CNN_F}$ and its residual-based variant $\mathrm{CNN_F}$_R. The main structures (e.g., the bed corner) contained in the guidance image are not well transferred to the target depth image, thereby resulting in blurry boundaries. In addition, inconsistent texture structures in the guidance image (e.g., the stripe pattern of the curtain on the wall) are also incorrectly copied to the target image. A potential approach that may improve the results is to adjust the architecture of $\mathrm{CNN_F}$, such as increasing the network depth or using larger filter sizes. However, as shown in Fig. 3e, these variants do not show notable improvement. Blurry boundaries and the texture-copying problem still occur. We note that similar observations have also been reported in [38], which indicate that the effectiveness of deeper structures for low-level tasks is not as apparent as that shown in high-level tasks (e.g., image classification).

We attribute the limitation of using a generic network for joint filtering to the fact that the original RGB guidance image fails to provide direct and effective guidance as it mixes a variety of information (e.g., texture, intensity, and edges). To validate this intuition, we show in Fig. 4 one example where we replace the original RGB guidance image with its edge map extracted using [39]. Compared to the results guided by the RGB image (Fig. 4d), the upsampled image using the edge map guidance (Fig. 4e) shows substantial improvement in preserving the sharp edges.

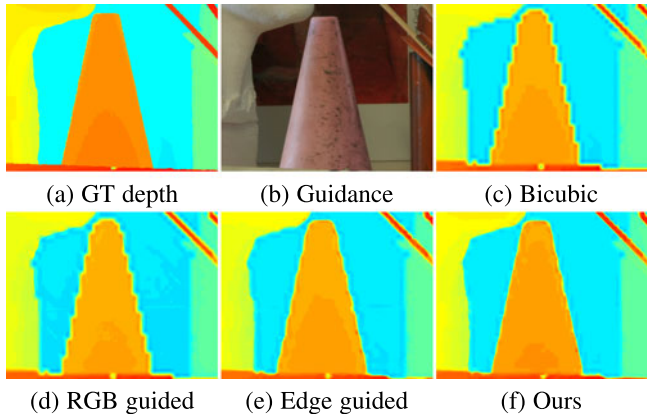


Fig. 4. *Comparison of different types of guidance.* Joint depth upsampling (8×) results using different types of guidance images. Both (d) and (e) are trained using the $\mathrm{CNN_F}$ network. Our method generates sharper boundary of the sculpture (left) and the cone (middle).

Based on the above observation, we introduce two subnetworks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ to first construct two separate processing streams for the two images before concatenation. With the proposed architecture, we constrain the network to extract effective features from both images separately first and then fuse them at a later stage to generate the final filtering output. This differs from conventional joint filters where the guidance information is mainly computed from the pixel-level intensity/color differences in the local neighborhoods. As our models are jointly trained in an end-to-end fashion, our result (Fig. 4f) shows further improvements over that of using the edge guided filtering (Fig. 4e).

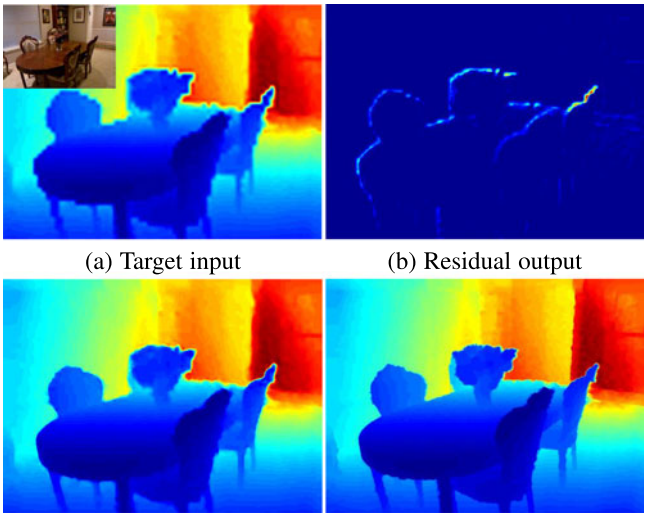
In this work, we adopt a three-layer structure for each sub-network as shown in Fig. 2. Given M training image samples $\{I_i^T, I_i^G, I_i^{gt}\}_{i=1}^M$, we learn the network parameters by minimizing the sum of the squared losses

$$\|I^{gt} - \Phi(I^T,\ I^G)\|_2^2\ , \tag{1}$$

where Φ denotes the joint image filtering operator. In addition, I^T, I^G, and I^{gt} denote the target image, the guidance image and the ground truth output, respectively.

3.2 Skip Connection

As the goal of the joint image filter is to leverage the signals from the guidance image to enhance the degraded target image, the input target image and the desired output share the same low-resolution frequency components. We thus introduce a skip connection to enforce the network to focus on learning the residuals rather than predicting the actual pixel values. With the skip connection, the network does not need to learn the identity mapping function from the input target image to the desired output in order to preserve the low-frequency contents. Instead, the network learns to predict the sparse residuals in important regions (e.g., object contours). In Fig. 5, we show an example of the predicted residuals, which highlights the estimated difference between the target input (Fig. 5a) and the ground truth (Fig. 5d). Quantitative results in Table 1 show that with the skip connection, the proposed algorithm obtains notable improvements over [18].



(a) Target input (b) Residual output

(c) Filtering output (d) Ground truth

Fig. 5. *Residual prediction.* Joint depth upsampling results (8×) of using our network with a skip connection. The filtering output (c) is the summation of (a) the target input and (b) the predicted output.

3.3 Network Training

Since the target and guidance image pair can be from various modalities (e.g., RGB/depth, RGB/NIR), it is infeasible and costly to collect large datasets and train one network for each type of data pair separately. The goal of our network training, however, is not predicting specific pixel values in one particular modality. Instead, we aim to train the network so that it can selectively transfer structures by leveraging the prior from the guidance image. Hence, we only need to train the network with only one type of image data and then apply the network to other domains.

To demonstrate that the proposed method is insensitive to the training data modality, we train the network with either the RGB/depth dataset [14] or RGB/flow dataset [40]. We conduct a cross-dataset evaluation (training with one type and evaluate on the other) and show the exemplary results in Fig. 6. Figs. 6a, 6b, 6c, and 6d shows the upsampled depth maps using models trained with different domains of image data. The flow model refers to the one trained with RGB/flow data for flow map upsampling, while the depth model is trained with RGB/depth data for depth map upsampling. In Fig. 6c, we apply the flow model to upsample the degraded depth map and show competitive results obtained by the depth model (Fig. 6d). Similar observations on flow map upsampling are also found in Figs. 6e, 6f, 6g, and 6h. Both the models trained with the flow and depth data achieve similar performance. More filtering results are shown in Section 4, where we evaluate the model with different image data from various domains. More quantitative results are presented in Table 1.

3.4 What Has the Network Learned?

Selective Transferring. Using the learned guidance model $\mathrm{CNN_G}$ alone to transfer details may sometimes be erroneous. In particular, the structures extracted from the guidance image may not exist in the target image. The top and middle rows of Fig. 7 show typical responses at the first layer of $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$. These two sub-networks show strong responses to edges from the target and guidance images respectively. Note that there are inconsistent structures in the guidance and target images, e.g., the window on the wall. The bottom row of Fig. 7 shows sample responses at the second layer of $\mathrm{CNN_F}$. We observe that the sub-network $\mathrm{CNN_F}$ suppresses inconsistent details.

We present another example in Fig. 8. We note that the ground truth depth map of the selected region is smooth. However, due to the high contrast patterns on the mat in the guidance image, several methods, e.g., [2], [4], incorrectly transfer the mat structure to the upsampled depth map. The reason is that these methods [2], [4] rely only on structures in the guidance image. The problem, commonly known as texture-copying artifacts, often occurs when the texture in the guidance image has strong color contrast. With the help of the $\mathrm{CNN_F}$, our method successfully blocks the texture structure in the guidance image (Fig. 8f).

Output of $\mathrm{CNN_G}$. In Fig. 9c, we show the learned guidance from $\mathrm{CNN_G}$ using two examples from the NYU v2 dataset [14]. In general, the learned guidance appears to be

TABLE 1
Quantitative Comparisons on Depth Upsampling

	Middlebury [16], [17]			NYU v2 [14]			SUN RGB-D [15]		
	4×	8×	16×	4×	8×	16×	4×	8×	16×
Bicubic	4.44 ± 1.59	7.58 ± 2.69	11.87 ± 4.04	8.16 ± 4.37	14.22 ± 7.56	22.32 ± 11.68	2.09 ± 1.56	3.45 ± 2.23	5.48 ± 3.21
MRF [22]	4.26 ± 1.52	7.43 ± 2.63	11.80 ± 4.01	7.84 ± 4.20	13.98 ± 7.42	22.20 ± 11.61	1.99 ± 1.57	3.38 ± 2.19	5.45 ± 3.18
GF [6]	4.01 ± 1.42	7.22 ± 2.55	11.70 ± 3.97	7.32 ± 3.86	13.62 ± 7.20	22.03 ± 11.51	1.91 ± 1.43	3.31 ± 2.15	5.41 ± 3.17
JBU [4]	2.44 ± 0.86	3.81 ± 1.49	6.13 ± 2.34	4.07 ± 2.22	8.29 ± 4.47	13.35 ± 7.47	1.37 ± 1.12	2.01 ± 1.76	3.15 ± 2.58
TGV [3]	3.39 ± 1.25	5.41 ± 1.99	12.03 ± 4.17	6.98 ± 3.61	11.23 ± 5.46	28.13 ± 10.47	1.94 ± 1.31	3.01 ± 2.46	5.87 ± 3.46
Park [2]	2.82 ± 0.94	4.08 ± 1.43	7.26 ± 2.41	5.21 ± 2.64	9.56 ± 4.41	18.10 ± 8.29	1.78 ± 1.33	2.76 ± 1.99	4.77 ± 2.97
Ham [13]	3.14 ± 1.24	5.03 ± 2.08	8.83 ± 3.96	5.27 ± 2.86	12.31 ± 6.07	19.24 ± 9.64	1.67 ± 1.41	2.60 ± 2.31	4.36 ± 3.32
DMSG [19]	**1.79 ± 0.66**	**3.39 ± 1.28**	**5.87 ± 2.38**	3.48 ± 1.96	6.07 ± 3.26	10.27 ± 5.79	1.30 ± 1.12	1.80 ± 1.31	2.81 ± 1.92
FBS [20]	2.58 ± 0.88	4.19 ± 1.48	7.30 ± 2.49	4.29 ± 2.53	8.94 ± 4.68	14.59 ± 8.32	1.58 ± 1.41	2.27 ± 2.33	3.76 ± 3.01
Ours-flow	2.31 ± 0.84	3.95 ± 1.45	6.34 ± 2.44	4.42 ± 2.77	7.32 ± 3.91	11.62 ± 6.58	1.36 ± 1.14	1.91 ± 1.37	2.90 ± 2.04
DJF[18]	2.14 ± 0.69	3.77 ± 1.32	6.12 ± 2.19	3.54 ± 1.86	6.20 ± 3.26	10.21 ± 5.57	1.28 ± 1.02	1.81 ± 1.35	2.78 ± 1.93
Ours	1.98 ± 0.67	3.61 ± 1.39	6.07 ± 2.20	**3.38 ± 1.95**	**5.86 ± 3.14**	**10.11 ± 5.49**	**1.27 ± 0.98**	**1.77 ± 1.30**	**2.75 ± 1.94**

Comparisons with the state-of-the-art methods in terms of RMSE. The depth values are scaled to the range $[0, 255]$ *for the Middlebury [16], [17], and SUN RGB-D[15] datasets. For the NYU v2 dataset [14], the depth values are measured in* centimeter. *Note that the depth maps in the SUN RGB-D dataset may contain missing regions due to the limitation of depth sensors. We ignore these pixels in calculating the RMSE. Numbers in bold indicate the best performance and underscored numbers indicate the second best. The mean and standard deviation of the RMSE values are shown in each entry.*

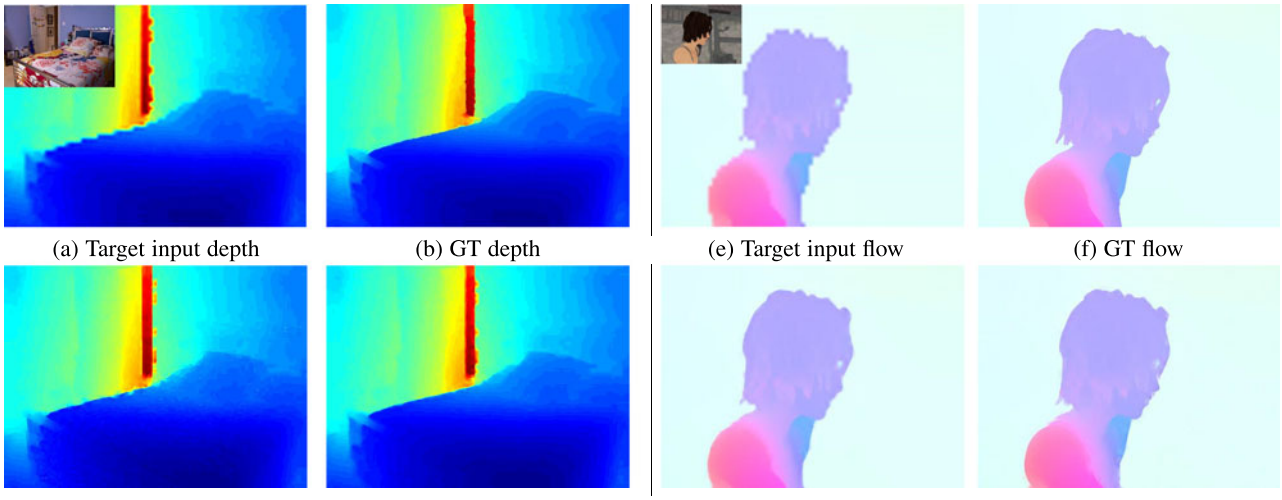


Fig. 6. *Effect of training data modalities.* (a)-(d) Joint depth map upsampling (8×). The model trained with RGB/flow data generates similar results when compared with the model trained with RGB/depth data. (e)-(h) Joint flow map upsampling (8×). (g) The model trained with RGB/depth data and (h) The model trained with RGB/flow data. The numbers are the RMSE metric comparing against the GT.

similar to an edge map highlighting the salient structures in the guidance image. We show edge detection results from [39] in Fig. 9d. Both results show strong responses to the main structures, but the guidance map generated by $\mathrm{CNN_G}$ appears to detect sharper boundaries while suppressing responses to small-scale textures, e.g., the wall in the first example. The result suggests that using only $\mathrm{CNN_F}$ (Fig. 3c) does not perform well due to lack of the salient feature extraction step from the sub-network $\mathrm{CNN_G}$.

To demonstrate the effectiveness of the skip connection, we compare the learned guidance without and with the skip connection in Figs. 9b and 9c. Adding the skip connection helps suppress more inconsistent structures (e.g., edges on the bed, wall, table) in the target/guidance pair, and consequently the residual-based model effectively alleviates texture-copying artifacts.

3.5 Relationship to Prior Work

The proposed framework is closely related to weighted-average, optimization-based, and CNN-based models. In each layer of the network, the convolutional filters also perform the weighted-average process. In this context, our filter is similar to the weighted-average filters. The key difference is that the weights in this work are learned from data while those of the weighted-average filters [4], [10] are predefined based on color or gradient features. The proposed network plays a similar role in the fidelity and regularization terms defined in the optimization-based joint filters. Specifically, the training objective in (1) corresponds to the fidelity term of the weighted-average filters [4], [10] as it encourages the output to be as close to the ground truth as possible. The skip connection implicitly serves as the regularization term by enforcing adjacent pixels to share similar

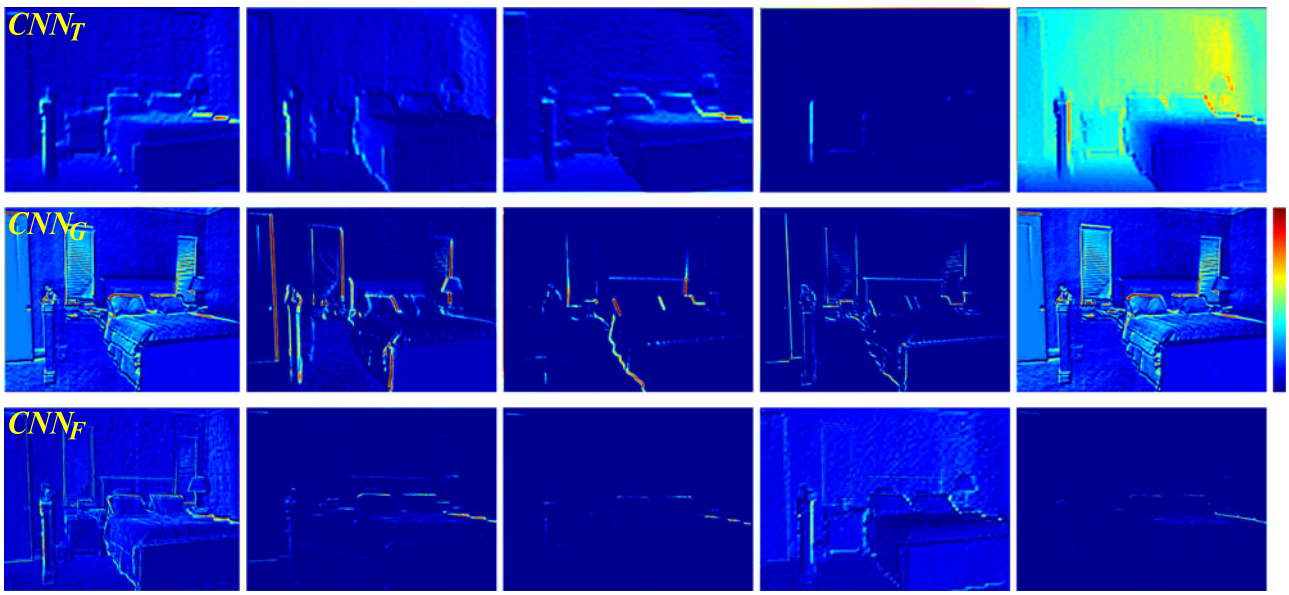
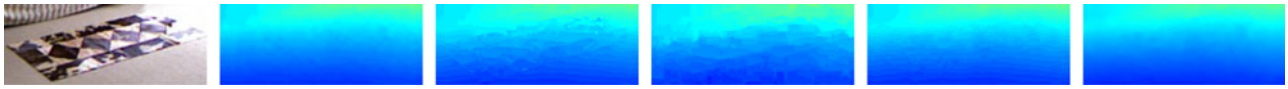


Fig. 7. *Visualization of feature responses.* Sample feature responses of the input in Fig. 9(a) at the first layer of $\mathrm{CNN_T}$ (top) and $\mathrm{CNN_G}$ (middle), and the second layer of $\mathrm{CNN_F}$ (bottom). For each subnetwork, we select five feature channels and visualize the responses through the colormap. The corresponding colorbar is shown in the rightmost. Note that with the help of $\mathrm{CNN_F}$, inconsistent structures (e.g., the window on the wall) are correctly suppressed.



(a) Guidance | (b) Ground truth RMSE | (c) JBU [4] 3.64 | (d) Park [2] 4.84 | (e) DJF [18] 3.24 | (f) Ours **2.95**

Fig. 8. *Selective transfer.* Comparisons of different joint upsampling methods on handling the texture-copying issue. The carpet on the floor contains grid-like texture structures that may be incorrectly transferred to the target image. The numbers are the RMSE metric comparing against the GT.

values (e.g., depth) as it directly bypasses the low-quality target input to the output of the network. For CNN-based models, our network architecture can be viewed as a unified model for different tasks. For example, if we remove $\mathrm{CNN_G}$ and use only $\mathrm{CNN_T}$ and $\mathrm{CNN_F}$, the resulting network architecture resembles an image restoration model, e.g., SRCNN [30]. On the other hand, in cases of removing $\mathrm{CNN_T}$, the remaining $\mathrm{CNN_G}$ and $\mathrm{CNN_F}$ can be viewed as one using CNNs for depth prediction [41].

4 Experimental Results

In this section, we demonstrate the effectiveness and applicability of our approach through a broad range of joint image filtering tasks, including joint image upsampling, texture-structure separation, and cross-modality image restoration. The source code and datasets will be made available to the public. More results can be found at http://vllab1.ucmerced.edu/~yli62/DJF_residual/.

Network Training. To train our network, we randomly collect 160,000 training patch pairs of 32×32 pixels from 1,000 RGB and depth images in the NYU v2 dataset [14]. Images in the NYU dataset are absolute depth maps captured in complicated indoor scenarios. We train two models for two different tasks: (1) joint image upsampling and (2) noise reduction. For the upsampling task, we obtain each low-quality target image from downsampling the ground-truth image (with scale factors of $4\times$, $8\times$, $16\times$) using the nearest neighbor interpolation. For the noise reduction task, we generate the low-quality target image by adding Gaussian noise to each of the ground-truth depth maps with zero mean and variance of 1e-3. We use the MatConvNet toolbox [42] to train our joint filters.

Testing. Using RGB/depth data for training, our model takes a 1-channel target image (depth map) and a 3-channel guidance image (RGB) as inputs. However, the trained model can be applied to other data types in addition to RGB/depth images with simple modifications. For the multi-channel target images, we apply the trained model independently for each channel. For the single-channel guidance images, we replicate it three times to create the 3-channel guidance image.

4.1 Depth Map Upsampling

Datasets. We present quantitative performance evaluation on joint depth upsampling using three benchmark datasets where the corresponding high-resolution RGB images are available:

- Middlebury dataset [16], [17]: We collect 30 images from 2001-2006 datasets with the missing depth values provided by Lu et al. [43].
- NYU v2 dataset [14]: As we use the 1,000 images in this dataset for training, we use the rest of 449 images for testing.
- SUN RGB-D [15]: We use a random subset of 2,000 high-quality RGB/depth image pairs from the 3,784 pairs captured by the Kinect v2 sensor. These images are captured from a variety of complicated indoor scenes.

(a) Guidance (inset: GT depth) | (b) Learned guidance w/o skip connection [18] | (c) Our learned guidance w/ skip connection | (d) Edge map [39]

Fig. 9. *Visualization of the learned guidance map.* Comparison between the learned guidance feature maps from $\mathrm{CNN_G}$ and edge maps from [39]. The network $\mathrm{CNN_G}$ is capable of extracting informative, salient structures from the guidance image for content transfer. Furthermore, with the skip connection, the learned guidance maps in (c) are cleaner than that in (b) by suppressing inconsistent structures (edges on the window and wall) in the target/guidance pair.

TABLE 2
Run-Time Performance Comparisons

	MRF [22]	GF [6]	JBU [4]	TGV [3]	Park [2]	Ham [13]	DMSG [19]	FBS [20]	Ours (CPU)	Ours (GPU)
Time (s)	0.76	0.08	5.64	68.21	45.79	8.62	0.71	0.34	1.31	0.07

Average run-time of depth map upsampling algorithms on images of size 640×480 *pixels.*

Note that the data in [14], [15] are *absolute* depth maps representing the physical distances in meters to the observer. However, the data in [16], [17] are *relative* depth maps (disparity), which measure the distance between two corresponding points in a scene under two different views. Each disparity value denotes the number of shifted pixels.

Evaluated Methods. We compare our model against several state-of-the-art joint image filters for depth map upsampling. The JBU [4], GF [6], Ham [13] and FBS [20] methods are generic joint image upsampling. On the other hand, the MRF [22], TGV [3], Park [2] and DMSG [19], algorithms are designed specifically for image-guided depth upsampling. Using the experimental protocols for evaluating the joint depth upsampling algorithms [2], [3], [13], we obtain the low-resolution target image from the ground-truth depth map using the nearest-neighbor downsampling method.

Quantitative Comparisons. Table 1 shows the quantitative results in terms of the root mean squared errors (RMSE). For other methods, we use the default parameters in the original implementations. The proposed algorithm performs well against the state-of-the-art methods across all three datasets. The extensive evaluations on absolute depth datasets [14], [15] demonstrate the effectiveness of our algorithm in handling complicated real-world indoor scenes. Furthermore, we compare the average run-time of different methods on the NYU v2 dataset in Table 2. We carry out all the experiments on the same machine with an Intel i7 3.6 GHz CPU and 16 GB RAM. We report the running time of our model in either CPU or GPU mode (GTX 745). Among all the evaluated methods, the proposed algorithm is efficient while delivering high-quality upsampling results.

The concurrent DMSG method by Tai et al. [19] outperforms the proposed algorithm on the Middlebury dataset. This can be attributed to several reasons. First, Tai et al. [19] leverage multi-scale guidance data while we use only single scale signals. The multi-scale design requires more network parameters to learn. For example, the model size of the upsampling model ($8\times$) in [19] is 1,822 KB compared to our model size of 526 KB. Second, the model in [19] is trained on a small collection of relative depth maps (82 images) [16], [17]. In contrast, our model is trained on a large dataset (1,000 images) of absolute depth maps [14]. For fair comparisons using absolute depth maps, we re-train the model of [19] with the same dataset [14] based on our own implementation. Table 1 shows that the performance of both [19] and our previous work DJF [18] on absolute depth datasets [14], [15] achieve similar performance. While the method in [19] also uses the similar strategy of predicting residuals, we demonstrate that the proposed algorithm achieves improved results with fewer parameters, suggesting the practical applicability of our model to real-world applications. Another important difference is that the model in [19] is designed only for depth upsampling. Our approach, on the other hand, can be applied to generic joint image filtering tasks.

Effects of Skip Connection. We validate the contribution of the introduced skip connection by comparing the DJF [18] method and proposed algorithm (bottom two rows of Table 1). In Section 5, we show that it is difficult to gain further improvement by simply modifying network parameters, such as the filter size, filter number, and network depth. However, with the skip connection, the proposed algorithm obtains significant performance improvement. The performance gain can be explained by that using skip connection alleviates the issues that the network only learns the appearance of the target input images, and helps the network focus on learning the residuals instead.

Effects of Training Modality. To validate the effect of training with different modalities, we compare our model with a variant that is trained with RGB/flow data (denoted as Ours-flow). We randomly select 1,000 RGB/flow image pairs from the Sintel dataset [40] and collect 80,000 training patch pairs of $3\,2\times 32$ pixels. We use either x-component or y-component of the optical flow as our target image. During the testing phase, we apply the trained model independently for each channel of the target image. Although the model Ours-flow is trained with the RGB/flow data for optical flow upsampling, Ours-flow performs favorably on the task of depth upsampling against our final model (Ours) trained with the RGB/depth data, as shown in Table 1.

Visual Comparisons. We show four examples for qualitative comparisons in Fig. 10. It is worth noticing that the proposed joint filter selectively transfers salient structures in the guidance image while avoiding texture-copying artifacts (see the green boxes). The GF [6] method does not recover the degraded boundary well under a large upsampling factor (e.g., $8\times$). The JBU [4], TGV [3] and Park [2] approaches are agnostic to structural consistency between the target and guidance images, and thus transfer erroneous details. In contrast, the results of our algorithm are smoother, sharper and more accurate with respect to the ground truth.

4.2 Joint Image Upsampling

Numerous computational photography applications require obtaining a solution map (e.g., chromaticity, saliency, disparity, labels) over the pixel grid. However, it is often time-consuming or memory-intensive to compute the high-resolution solution maps directly. An alternative is to first obtain a low-res solution map over the downsampled pixel grids and then upsample the low-resolution solution map back to the original resolution with a joint image upsampling algorithm. Such a pipeline requires the upsampling method to restore well image degradation caused by downsampling and avoid the inconsistency issues. In what follows, we demonstrate the use of the learned joint image filters for colorization and saliency as examples. Note that in the following applications we use the *same* model trained with RGB/depth data and evaluate on other image modalities without retraining the network using data in the new domains.

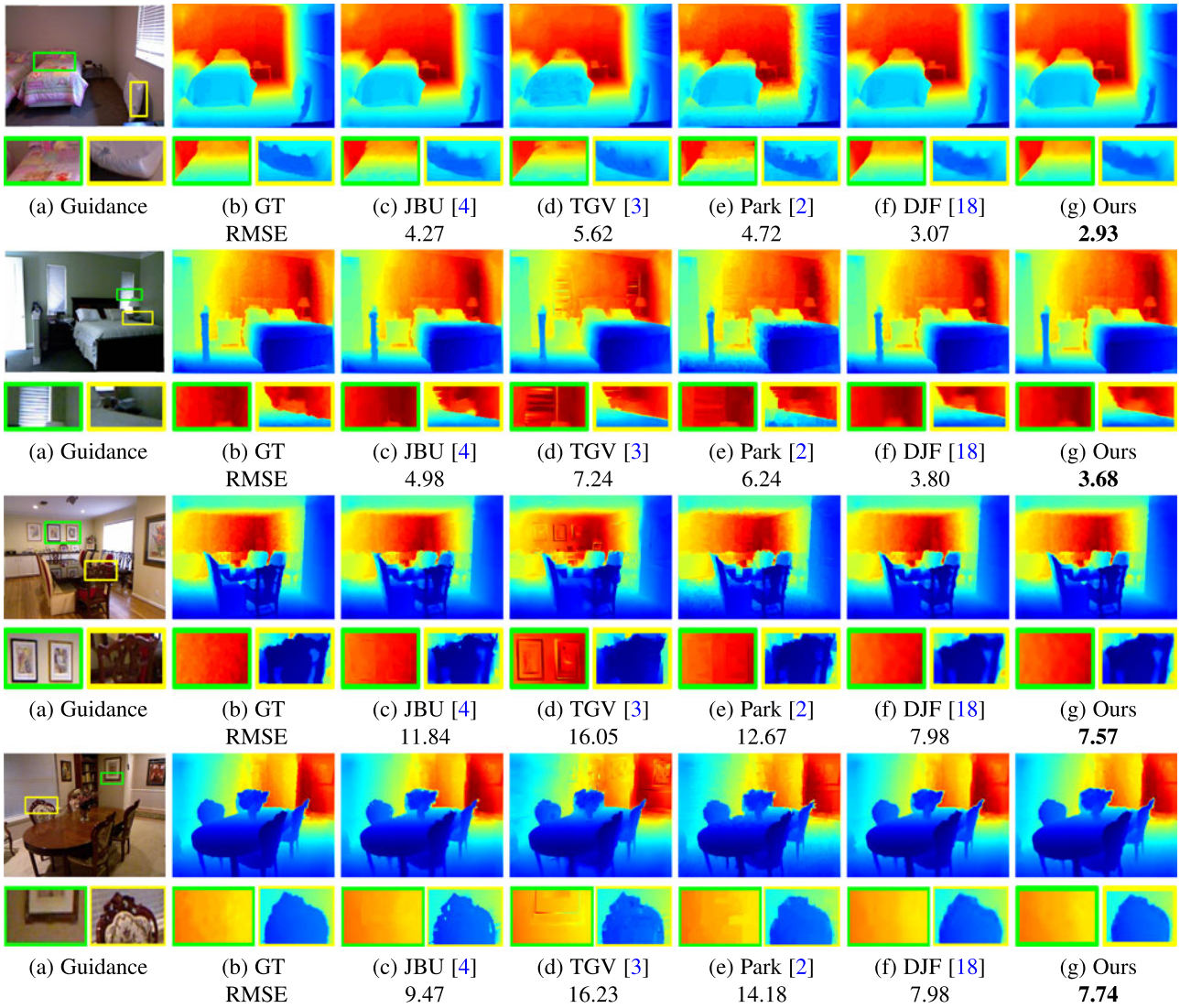


Fig. 10. *Qualitative comparisons on depth upsampling*. Comparisons against existing depth upsampling algorithms for a scaling factor of 8×. The numbers (in centimeter) are the RMSE metric comparing against the GT in (b).

For the colorization task, we first compute the chromaticity map on the downsampled (4×) image using the user-specified color scribbles [44]. We then use the original high-resolution intensity image as the guidance image to jointly upsample the low-resolution chromaticity map. Fig. 11 shows that our model is able to achieve visually pleasing results with fewer color bleeding artifacts and efficiently. Our results are visually similar to the direct solutions on the high-resolution intensity images (Fig. 11b). The quantitative comparisons are presented in the first row of Table 3. We use the direct solution of [44] on the high-resolution image as ground truth and compute the RMSE over seven test images in [44]. Table 3 shows that our method performs well with the lowest error. Note that our pipeline (low-res result + joint upsampling) is nearly three times faster (2.82 seconds) than directly running the colorization algorithm [44] on the original pixel grid to obtain the high-resolution result (8.20 seconds). Note that for fair comparisons, all run-time results are obtained based on the CPU mode.

For saliency detection, we first compute the saliency map on the downsampled (10×) image using the manifold method by Yang et al. [45]. We then use the original high-resolution intensity image as guidance to upsample the low-resolution saliency map. Fig. 12 shows the saliency detection results by the state-of-the-art methods and proposed algorithm. Overall, the proposed algorithm generates sharper edges than other alternatives. In addition, we present quantitative evaluation using the ASD benchmark dataset [47] which consists of 1,000 images with manually labeled ground truth. Table 3 shows the comparison between different upsampling methods and our approach in terms of F-measure [48]. The experimental results demonstrate that the proposed algorithm performs favorably against the state-of-the-art methods.

4.3 Structure-Texture Separation

We apply our model trained for noise reduction to the task of structure-texture separation. Here we use the target image itself as the guidance. We adopt a similar strategy as in the rolling guidance filter (RGF) [9] to remove small-scale textures, i.e., using the output of the previous iteration as the input of the current iteration.

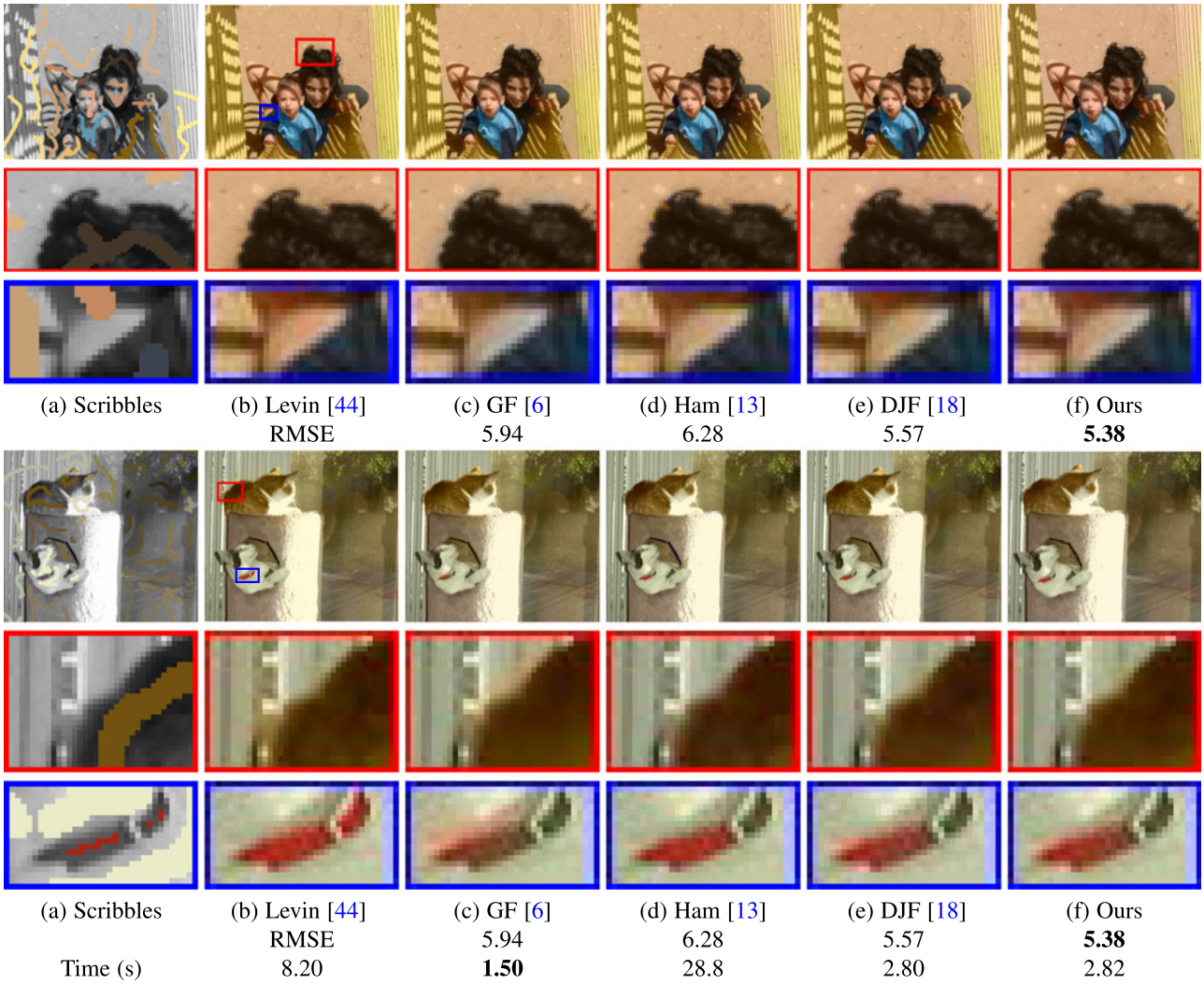


Fig. 11. *Colorization upsampling.* Joint image upsampling applied to colorization. We also list the runtime for the colorization upsampling process for each method. The close-up areas show that our joint upsampling results (f) have fewer color bleeding artifacts when compared with other competing algorithms (c-e). Our visual results (f) are comparable with the results computed using the full resolution image in (b). The RMSE metric comparing against the GT in (b) are presented. The average RMSE over all test images are shown in Table 3.

We use the inverse halftoning task as an example. A halftoned image is generated by the reprographic technique that simulates continuous tone imagery using various dot patterns [46], as shown in Fig. 13a. The goal of inverse halftoning is to remove these dots while preserving the main structures. We compare our results with those from the RGF [9], Xu [8], DJF [18] and the method by Kopf [46] for halftoned images reconstruction. Since there exists no ground truth data, we use the results from Kopf [46] as the pseudo ground truth as it is specifically designed for reconstructing halftoned images and achieves the best visual quality. For [8], [9], we carefully select the parameters (listed in Fig. 13) for the optimal results by considering both removing the dot patterns and keeping the sharp edges intact. We use the same high-resolution test images from [46] and present two zoomed-in patch examples in Fig. 13 for illustration, where one (top) is with small-scale dots and another one (bottom) is with large-scale dots. For the DJF [18] and proposed method, we show the results of running two iterations in the first row and three iterations in the second row of Figs. 13e and 13f. Our model achieves better results on removing small-scale dots but worse results on removing large-scale dots compared with the methods in [8], [9]. However, in order to get the best results, both [8], [9] require to manually select optimal parameters for different inputs. Our model (trained on RGB/depth only) is not expected to consistently achieve the best performance but able to generalize well for comparable results on the inverse halftoning task without tuning parameters.

TABLE 3
Quantitative Comparisons of Different Upsampling Methods on Difference Solution Maps

	Bicubic	GF [6]	Ham [13]	DJF [18]	Ours
RMSE	6.01	5.74	6.31	5.48	5.40
F-measure	0.759	0.766	0.763	0.778	0.781

4.4 Cross-Modality Filtering for Noise Reduction

Here, we demonstrate that our model can handle various visual domains through two noise reduction applications using RGB/NIR and flash/non-flash image pairs. Fig. 14 (left) show sample results on joint image denoising with the

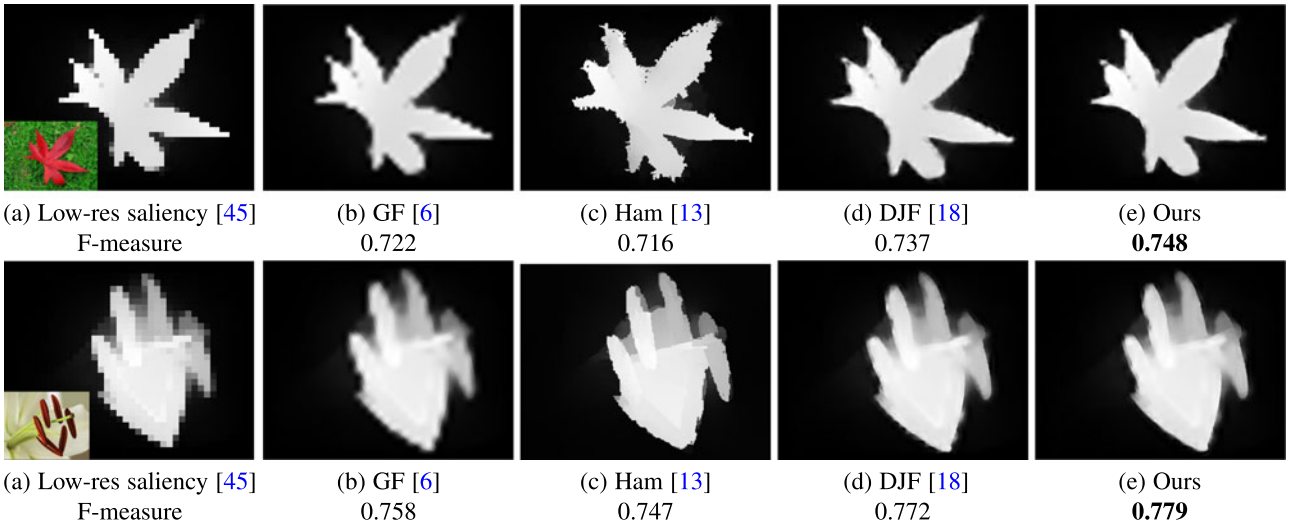


Fig. 12. *Saliency map upsampling*. Visual comparisons of saliency map upsampling results (10×). (a) Low-res saliency map obtained from the downsampled RGB image (inset: guidance image). The numbers are the F-measure metric comparing against the GT. The average F-measure over all test images are shown in Table 3.

NIR guidance image. The filtering results by our method are comparable to those of the state-of-the-art technique [5]. For flash/non-flash image pairs, we aim to merge the ambient qualities of the no-flash image with the high-frequency details of the flash image. Guided by a flash image, the filtering result of our method is comparable to that of [5], as shown in Fig. 14 (right).

5 Discussions

In this section, we first analyze the effects of the performance under different hyper-parameter settings using the network architecture in Fig. 2. Then, we discuss several limitations of the proposed algorithm. To validate the design choices, we vary the filter number n, filter size f, and depth d of each sub-network. We use the same training process as described in Section 4 and evaluate different models on the NYU v2 dataset [14] for 8× upsampling in terms of RMSE.

5.1 Filter Number

We first analyze the effects of the number of filters (n_1, n_2) in first two layers of each sub-network. The quantitative results are shown in Table 4. In the setting of without the skip connection (top row), we observe that larger filter number may not always result in performance improvements because it increases the difficulty of training the network. The results suggest that the performance of such network design is somewhat saturated with the sufficient number of filters. In order to get further improvements, we need to adjust the network design or the learning objectives, rather than simply modifying hyper-parameters.

Such a hypothesis is supported by the setting of with the skip connection, where we add a skip connection to the entire network and reformulate the network as learning residual functions. The bottom row of Table 4 shows that the filter number do yield progressive improvements when it is increased. This is in accordance with the observation in [34], [36] where residual learning is more effective for training the network with larger capacity. However, a larger network also slows down the training process and may only provide marginal performance improvements. Consequently, the selected hyper-parameters of our method (shown in Fig. 2) strike a good balance between accuracy and computational efficiency.

Furthermore, we discuss the effects of the output channels (n_3) of $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ and show the results in Table 5. Intuitively, using multi-dimensional features may improve the model capacity and therefore its performance.

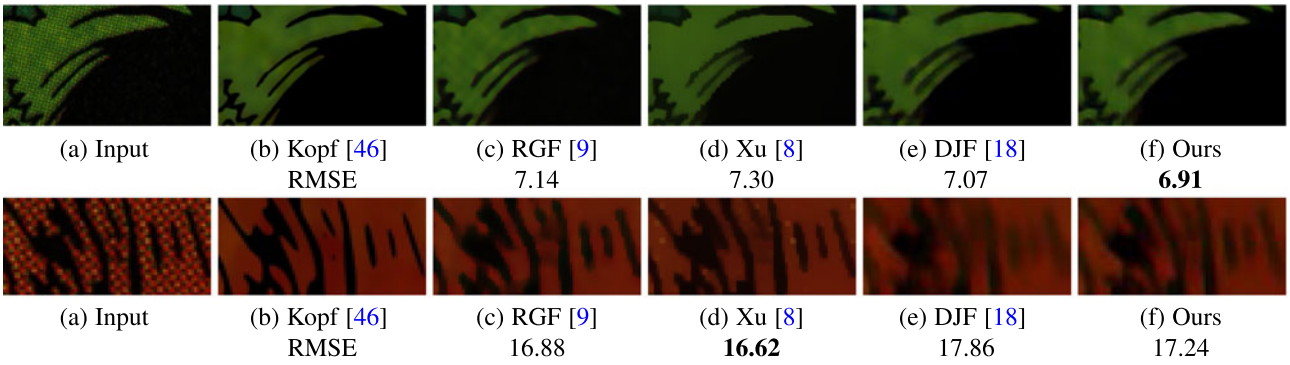


Fig. 13. *Inverse halftoning*. For each method, we carefully select the parameter for the optimal results. (c) $\sigma_s = 2, \sigma_r = 0.05, iter = 4$. (d) $\lambda = 0.005, \sigma = 1$. (e)-(f) top: $iter = 2$, bottom: $iter = 3$. Since there exists no GT result, we regard the result of [46] in (b) as the GT because it is an algorithm specifically designed for reconstructing halftoned images. The numbers are the RMSE metric comparing against the result in (b).

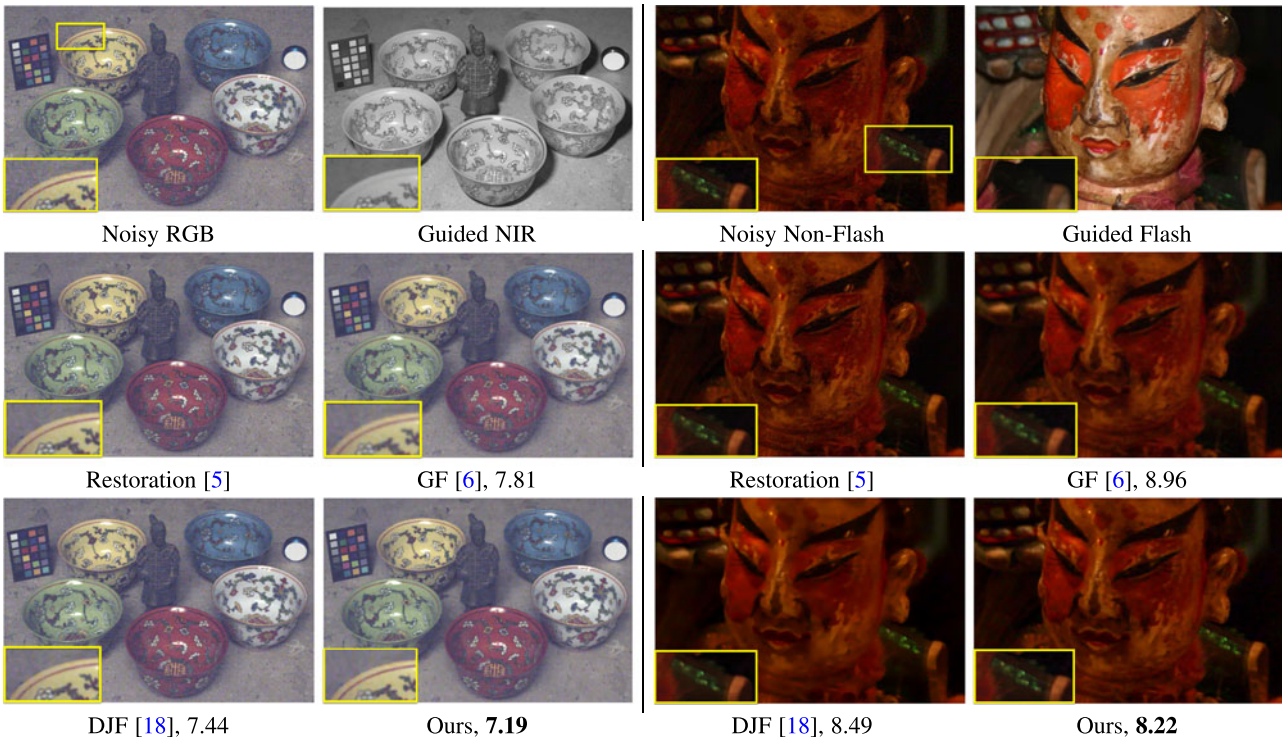


Fig. 14. *Cross-modality filtering for noise reduction.* Left: Results of noise reduction using RGB/NIR image pairs (Target: RGB, Guidance: NIR). Right: Results of noise reduction using flash/non-flash image pairs (Target: Non-Flash, Guidance: Flash). The numbers are the RMSE metric comparing against the result of [5].

However, our experimental results indicate that using multi-dimensional feature maps only slows down the training process without clear performance gain, for both without and with the skip connection settings. Therefore, we set the output feature maps extracted from the target and guidance images as one single channel ($n_3 = 1$).

5.2 Filter Size

We examine the network sensitivity to the spatial support of the filters. With all the other experimental settings kept the same, we gradually increase the filter size f_i (i = 1, 2, 3) in different layers and show the corresponding performance in Table 6.

TABLE 4
Quantitative Results (RMSE in Centimeters for 8×) of Using Different Filter *Numbers* in Each Sub-Network

$n_1 = 256$	$n_1 = 128$	$n_1 = 96$	$n_1 = 64$
$n_2 = 128$,	$n_2 = 64$	$n_2 = 48$	$n_2 = 32$
6.40	6.44	6.32	6.35
5.82	5.84	5.90	5.97

We apply the same parameters to three sub-networks. Top: Without the skip connection, Bottom: With the skip connection.

TABLE 5
Quantitative Results (RMSE in Centimeters for 8×) of Using Different Filter *Numbers* in the 3rd Layer of $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$

$n_3 = 1$	$n_3 = 16$	$n_3 = 32$	$n_3 = 64$
6.20	6.40	6.24	6.34
5.86	6.11	5.93	6.02

Top: Without the skip connection, Bottom: With the skip connection.

Starting from using small filter sizes ($f_1 = 5$, $f_2 = 1$, $f_3 = 3$), we observe a steady trend of improvements when increasing the filter sizes. This is because smaller filters will restrict the network to focus on detailed local smooth regions that provide little information for restoration. In contrast, a reasonably large filter size can cover richer structural cues that lead to better results. However, when we further enlarge the filter size (e.g., up to $f_1 = 11$, $f_2 = 3$, $f_3 = 7$), we do not see additional performance gain. We attribute this to the increasing difficulty of network training because larger filter sizes indicate more number of parameters to be learned. Consequently, we choose the filter size $f_1 = 9$, $f_2 = 1$, and $f_3 = 5$ as a good trade-off between the efficiency and performance.

5.3 Network Depth

As suggested in [38] that the number of layers does not play a significant role in non-residual based models for low-level tasks, we focus on evaluating the residual-based model (with the skip connection) with different network depth. First, we analyze whether using one generic but deeper

TABLE 6
Quantitative Results (RMSE in Centimeters for 8×) of Using Different Filter *Sizes* in Each Sub-Network

$f_1 = 11$	$f_1 = 9$	$f_1 = 9$	$f_1 = 7$	$f_1 = 5$
$f_2 = 3$	$f_2 = 3$	$f_2 = 1$	$f_2 = 1$	$f_2 = 1$
$f_3 = 7$	$f_3 = 7$	$f_3 = 5$	$f_3 = 5$	$f_3 = 3$
6.28	6.40	6.20	6.47	6.62
5.93	6.05	5.86	6.06	6.24

Top: Without the skip connection, Bottom: w/ the skip connection.

TABLE 7
Quantitative Evaluation (RMSE in Centimeters for 8×) When Using Residual-Based $\mathrm{CNN_F\_R}$ Only Under Different Network Depth d

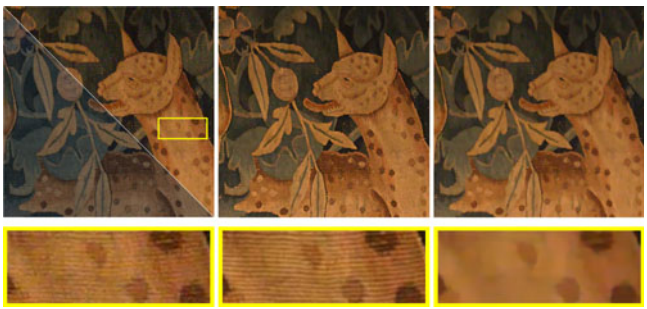
$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$	Ours
6.31	6.25	6.22	6.20	6.17	6.16	5.86

TABLE 8
Quantitative Evaluation of Our Model by Increasing the Number of Layers (the Depth d) Used in Each Subnetwork

	$d=2$	$d=3$	$d=4$	$d=5$
RMSE/cm	5.99	5.86	5.77	5.73
Model size/MB	0.48	0.53	5.0	11.4

TABLE 9
Quantitative Evaluation of Different Combinations of Network Depth of $\mathrm{CNN_T}$ ($\mathrm{CNN_G}$) and $\mathrm{CNN_F}$

$\mathrm{CNN_T/CNN_G}-\mathrm{CNN_F}$	$0/0-6$	$2/2-4$	$3/3-3$	$4/4-2$
RMSE/cm	6.13	5.95	5.86	6.03



(a) Input (b) Georg et al. [12] (c) Ours

Fig. 15. *Failure cases*. Detailed small-scale textures (yellow rectangle) in the guidance image are over-smoothed by our filter.

residual-based $\mathrm{CNN_F\_R}$ network can improve the performance. We gradually increase the depth from 3 to 8 and show the results in Table 7. Overall, the performance of the $\mathrm{CNN_F\_R}$ network improves with a deeper network. However, the performance quickly reaches the point of diminishing returns after d is larger than 4.

Next, we evaluate our model (three subnetworks) by increasing the network depth. We simultaneously increase the depth d of each subnetwork from 2 to 5 and show the corresponding results in Table 8. We observe that equipped with the skip connection a deeper network generally leads to better performance. This is in accordance with the observation in [36] where a 20-layer deep residual net is used for image super-resolution. However, in our case with three subnetworks, the deeper network also induces fast growth of model size as well as longer training time. We find the performance improvement is incremental when d is varied from 3 to 5. Thus, we set d to 3 as a trade-off between model size and performance.

5.4 Merging Layer

As shown in Fig. 2, the $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ are merged at the output (third) layer. Here we further analyze the effect of merging $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ at different layers. We fix the whole network depth as 6 and analyze different combinations of network depth of $\mathrm{CNN_T}$, $\mathrm{CNN_G}$ and $\mathrm{CNN_F}$. We gradually increase the depth of $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ while decreasing the depth of $\mathrm{CNN_F}$ (in order to maintain the overall network depth). For example, $0/0-6$ (Table 9) indicates that we directly stack the target and guidance image and applying a 6-layer $\mathrm{CNN_F}$ only. The evaluation results of different models are shown in Table 9. Overall, deeper target/guidance networks ($\mathrm{CNN_T}$ and $\mathrm{CNN_G}$) result in sizable performance improvement resulting from effective feature extractions. However, as the $\mathrm{CNN_F}$ becomes shallower, the performance degrades again. This indicates that neither the $\mathrm{CNN_T}$ ($\mathrm{CNN_G}$) nor the $\mathrm{CNN_F}$ should be too shallow. Therefore, we chose the combination of $3/3-3$ for best performance.

5.5 Limitations

We note that in some images, our model fails to transfer small-scale details from the guidance map. In such cases, our model incorrectly treats certain small-scale details as noise. This can be explained by the fact that our training data is based on depth images that are mostly smooth and does not contain many spatial details.

Fig. 15 shows two examples of a flash/non-flash pair for noise reduction. There are several spotty textures on the porcelain in the guided flash image that should have been preserved when filtering the noisy non-flash image. Similarly, our method is not able to effectively transfer the small-scale strip textures on the carpet to the target image. Compared with the method by Georg et al. [12] (Figs. 15b and 15d) that is designed specifically for flash/non-flash images, our filter treats these small-scale details as noise and tends to over-smooth the contents. We will collect more training data from other domains (e.g., flash/non-flash) to address the over-smoothing problem in our future work.

6 Conclusions

In this paper, we present a learning-based approach for joint filtering based on convolutional neural networks. Instead of relying only on the guidance image, we design two subnetworks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ to extract informative features from both the target and guidance images. These feature maps are then concatenated as inputs for the network $\mathrm{CNN_F}$ to selectively transfer salient structures from the guidance image to the target image while suppressing structures that are not consistent in both images. While we train our network on one type of data (RGB/depth or RGB/flow), our model generalizes well on handling images in various modalities, e.g., RGB/NIR and flash/non-Flash image pairs. We show that the proposed algorithm is computationally efficient and performs favorably against the state-of-the-art techniques on a wide variety of computer vision and computational photography applications, including cross-modal denoising, joint image upsampling, and texture-structure separation.

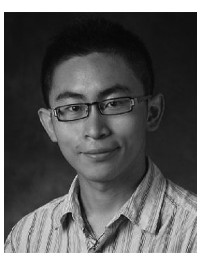
References

[1] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

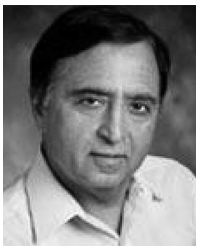
[2] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1623–1630.

[3] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 993–1000.
[4] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in *Proc. ACM SIGGRAPH*, 2007, Art. no. 96.
[5] Q. Yan, X. Shen, L. Xu, S. Zhuo, X. Zhang, L. Shen, and J. Jia, "Cross-field joint image restoration via scale map," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1537–1544.
[6] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
[7] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3406–3414.
[8] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, 2012, Art. no. 139.
[9] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 815–830.
[10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
[11] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," in *Proc. ACM SIGGRAPH*, 2004, pp. 673–678.
[12] P. Georg, A. Maneesh, H. Hugues, S. Richard, C. Michael, and T. Kentaro, "Digital photography with flash and no-flash image pairs," in *Proc. ACM SIGGRAPH*, 2004, pp. 664–672.
[13] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4823–4831.
[14] P. K. N. Silberman, D. Hoiem, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
[15] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.
[16] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[17] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[18] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 154–169.
[19] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 353–369.
[20] J. T. Barron and B. Poole, "The fast bilateral solver," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 617–632.
[21] M.-Y. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic upsampling of depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 169–176.
[22] J. Diebel and S. Thrun, "An application of Markov random fields to range sensing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 291–298.
[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[24] V. Jampani, M. Kiefel, and P. V. Gehler, "Learning sparse high dimensional filters: Image filtering, dense CRFs and bilateral neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4452–4461.
[25] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, 2017, Art. no. 118.
[26] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2516–2525.
[27] S. Gu, W. Zuo, S. Guo, Y. Chen, C. Chen, and L. Zhang, "Learning dynamic guidance for depth image enhancement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 712–721.
[28] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2392–2399.
[29] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 633–640.
[30] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.
[31] J. Zhang, J. Pan, W.-S. Lai, R. Lau, and M.-H. Yang, "Learning fully convolutional networks for iterative non-blind deconvolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6969–6977.
[32] F. Philipp, D. Alexey, I. Eddy, H. Philip, H. Caner, G. Vladimir, V. D. S. Patrick, C. Daniel, and B. Thomas, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2758–2766.
[33] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1669–1678.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[35] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
[36] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.
[37] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5835–5843.
[38] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
[39] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1841–1848.
[40] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 611–625.
[41] E. David, P. Christian, and F. Rob, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.
[42] V. Andrea and L. Karel, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 689–692.
[43] S. Lu, X. Ren, and F. Liu, "Depth enhancement via low-rank matrix completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3390–3397.
[44] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH*, 2004, pp. 689–694.
[45] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3166–3173.
[46] J. Kopf and D. Lischinski, "Digital reconstruction of halftoned color comics," *ACM Trans. Graph.*, vol. 31, 2012, Art. no. 140.
[47] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1597–1604.
[48] L. Mai and F. Liu, "Comparing salient object detection results without ground truth," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 76–91.

Yijun Li received the BS degree from Zhejiang University, in 2012, and the MS degree from Shanghai Jiao Tong University, in 2015. He is working toward the PhD degree in electrical engineering and computer science at the University of California, Merced. His research interests lie in the computer vision and machine learning, including image generation, synthesis, and low-level vision.



Jia-Bin Huang received the BS degree in electronics engineering from National Chiao-Tung University, Hsinchu, Taiwan, and the PhD degree from the Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign, in 2016. He is an assistant professor with the Bradley Department of Electrical and Computer Engineering, Virginia Tech. He is a member of the IEEE.



Narendra Ahuja received the PhD degree from the University of Maryland, College Park, Maryland, in 1979. He is the Donald Biggar Willet professor with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois. He is a fellow of the American Association for Artificial Intelligence, the International Association for Pattern Recognition, the Association for Computing Machinery, the American Association for the Advancement of Science, and the International Society for Optical Engineering.

Ming-Hsuan Yang received the PhD degree in computer science from the University of Illinois at Urbana-Champaign, in 2000. He is a professor in electrical engineering and computer science with the University of California, Merced. Prior to joining UC Merced in 2008, he was a senior research scientist with the Honda Research Institute working on vision problems related to humanoid robots. He served as an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* from 2007 to 2011, and is an associate editor of the *International Journal of Computer Vision*, the *Image and Vision Computing*, and the *Journal of Artificial Intelligence Research*. He received the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and ACM.