This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Dynamic Scene Deblurring Using Spatially Variant Recurrent Neural Networks

Jiawei Zhang[1,2*] Jinshan Pan[3†] Jimmy Ren[2] Yibing Song[4]
Linchao Bao[4] Rynson W.H. Lau[1] Ming-Hsuan Yang[5]
[1]Department of Computer Science, City University of Hong Kong [2]SenseTime Research
[3]School of Computer Science and Engineering, Nanjing University of Science and Technology
[4]Tencent AI Lab [5]Electrical Engineering and Computer Science, University of California, Merced

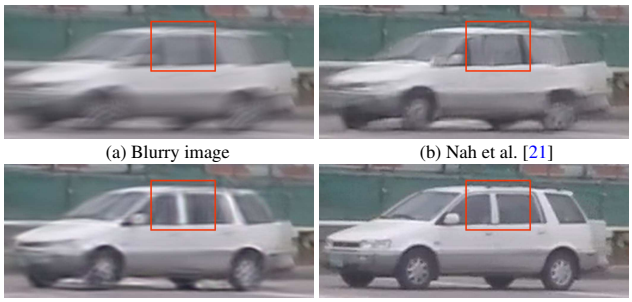
Abstract

Due to the spatially variant blur caused by camera shake and object motions under different scene depths, deblurring images captured from dynamic scenes is challenging. Although recent works based on deep neural networks have shown great progress on this problem, their models are usually large and computationally expensive. In this paper, we propose a novel spatially variant neural network to address the problem. The proposed network is composed of three deep convolutional neural networks (CNNs) and a recurrent neural network (RNN). RNN is used as a deconvolution operator performed on feature maps extracted from the input image by one of the CNNs. Another CNN is used to learn the weights for the RNN at every location. As a result, the RNN is spatially variant and could implicitly model the deblurring process with spatially variant kernels. The third CNN is used to reconstruct the final deblurred feature maps into restored image. The whole network is end-to-end trainable. Our analysis shows that the proposed network has a large receptive field even with a small model size. Quantitative and qualitative evaluations on public datasets demonstrate that the proposed method performs favorably against state-of-the-art algorithms in terms of accuracy, speed, and model size.

1. Introduction

Motion blur, which is caused by camera shake and object motions, is one of the most common problems when taking pictures. The community has made active research efforts on this classical problem in the last decade. However, restoring a clean image from blurry one is difficult since it is a highly ill-posed problem. Most existing algorithms assume the blur to be caused by camera motions, such as translation and rotation. However, this assumption does not always hold for dynamic scenes, which contain object motions and abrupt depth variations (e.g., Figure 1).

Existing dynamic scene deblurring algorithms [9, 10, 23] usually need segmentation methods to help the deblurring process. However, these methods heavily depend on an accurate segmentation. In addition, the deblurring process is time-consuming as highly non-convex optimization problems should be solved.



(a) Blurry image (b) Nah et al. [21] (c) Ours (d) Clean image

Figure 1. A challenging dynamic scene blurry example where the blur is caused by both camera shake and object motion. As the blur is spatially variant, conventional CNN-based methods (which usually adopt convolution and non-linear activation operations, e.g., Nah et al. [21] to approximate this problem) do not handle this problem well. Our method is based on a spatially variant RNNs, which is able to model the spatially variant property, capture a larger receptive field, and thus generate a much clearer image.

Recently, deep convolutional neural networks (CNNs) have been applied to dynamic scene deblurring [33, 7, 21, 22]. Unlike conventional algorithms that involve a complex blur kernel estimation process, these CNN-based methods either predict pixel-wise blur kernels or directly restore clear images from blurred inputs. However, existing CNN-based methods have two major problems. The first one is that weights of the CNN are spatially invariant. It is hard to use a CNN with a small model size to approximate the dynamic scene deblurring problem, which has the spatially variant property (see Figure 1). The second one is that large image regions should be used to increase the receptive field even though the blur is small. This inevitably leads to a network with a large model size and a high computation cost. Thus, there is a need to develop an effective network with a small model size and large receptive field to restore clear images from blurred dynamic scenes.

In this paper, we propose a spatially variant recurrent

*email: zhjw1988@gmail.com
†Corresponding author

neural network (RNN) for dynamic scene deblurring, where the pixel-wise weights of the RNN are learned by a deep CNN. In the CNN, the auto-encoder framework is proposed to reduce the model size of the proposed network and facilitate pixel-wise weight estimation. Our analysis shows that the RNN model can be regarded as a deconvolution operation and is able to model the spatially variant blur. The proposed network can be trained in an end-to-end manner.

The contributions of this paper are summarized as follows:

- We propose a novel end-to-end trainable spatially variant RNN for dynamic scene deblurring. The pixel-wise weights of the RNNs are learned by a deep CNN, which is able to facilitate the spatially variant blur removal.
- We show that the deblurring process can be formulated by an infinite impulse response (IIR) model. We further analyze the relationship between the proposed spatially variant RNN and the deblurring process, and show that the spatially variant RNN has a large receptive field and is able to model the deblurring process.
- We evaluate the proposed model on the benchmark datasets quantitatively and qualitatively and show that the proposed method performs favorably against state-of-the-art algorithms in terms of accuracy, speed as well as model size.

2. Related Work

Dynamic scene deblurring is a highly ill-posed problem. Conventional methods [9, 10, 23] usually add constrains on the estimated image and blur kernel, and then optimize complex objective functions. In [9], a segmentation-based algorithm is proposed to jointly estimate motion segments, the blur kernel, and the latent image. However, these methods cannot handle forward motions and depth variations. Kim et al. [10] propose a segmentation-free dynamic scene deblurring algorithm. This method assumes that the blur kernels can be modeled by a local linear optical flow field. This assumption does not always hold as real-world motions are complex. Pan et al. [23] propose an algorithm based on soft-segmentation. To handle large blur, this method introduces a segmentation confidence map into the conventional deblurring framework. However, it requires user inputs to initialize segmentations.

Recently, deep learning has been widely used in many low-level vision problems, such as denoising [1, 20, 45], super-resolution[4, 35, 13, 14, 15, 43, 31], dehazing [27], derain/dedirt [6, 5], edge-preserving filtering [39, 18], and image deblurring (non-blind [28, 38, 44] and blind [29, 2, 42]).

Several methods [33, 7] use deep learning to estimate the non-uniform blur kernel and then utilize a non-blind deblurring algorithm [46] to obtain sharp images in dynamic scene deblurring. Sun et al. [33] propose a deep CNN model to estimate the motion blur of every patch. The Markov random field (MRF) is then used to obtain a dense motion field. However, as the network is trained at the patch-level, it cannot fully utilize the high-level information from a larger region. Gong et al. [7] propose a deeper CNN to estimate the motion flow without post-processing. However, this method is only designed for linear blur kernels, which limits the application domains. In addition, the networks used in [33] and [7] are not trained in an end-to-end manner. The image restoration process requires a conventional non-blind deblurring step, e.g., [46], which is time-consuming.

Some deblurring algorithms based on end-to-end trainable neural networks have also been proposed [21, 22, 8, 34]. To use a large receptive field in the network for image restoration, most of these algorithms develop a multi-scale strategy or very deep models. Noroozi et al. [22] adopt skip connections. The network only needs to generate the residual image to reduce the difficulty of reconstruction. Nah et al. [21] propose a very deep residual network with 40 convolution layers in every scale, and a total of 120 convolution layers. The adversarial loss is used in their network to obtain sharp realistic results. In addition, since the blur varies from image to image and from pixel to pixel, it is inefficient to use the same network parameters to handle all cases. Some methods are designed for text or license plate deblurring [8, 34], and cannot be easily extended to handle dynamic scene deblurring.

We note that the aforementioned end-to-end networks need to have a very deep network structure [21] or a large number of channels [22]. Since blur is spatially variant in dynamic scenes, only using CNNs might be inefficient. In addition, it is difficult to use a single CNN model to deal with different blurs. For example, Xu et al. [38] propose a neural network for non-blind deblurring, but need to train different networks for different kernels.

Spatially variant neural networks [26, 19] have been developed for low-level vision tasks. For example, a shepard interpolation layer is proposed in [26] for inpainting and super-resolution. They use a predefined mask to indicate whether a pixel is used for interpolation to achieve spatially variant operation. A spatially variant RNN is proposed in [19], where spatially-variant weights of the RNN is learned by a deep CNN. By utilizing spatially variant RNN, the network in [19] does not need to use a large number of channels or large kernels since image information can be propagated for a long distance by the RNN. As the blur in dynamic scene deblurring is spatially variant, we need to involve both a large region and a spatially variant structure. To this end, we propose a novel spatially variant RNN based on an end-to-end trainable network.

3. Proposed Method

In this section, we show that the deconvolution/deblurring step is equivalent to an infinite impulse response (IIR) model [25], which can be approximated by RNNs. We then present the structure of the spatially variant RNN for dynamic scene deblurring, where the pixel-wise weights of the spatially variant RNN are learned by a deep CNN.

3.1. Motivation

Given a 1D signal x and a blur kernel k, the blur process can be formulated as:

$$y[n] = \sum_{m=0}^{M} k[m]x[n-m], \tag{1}$$

where y is the blurred signal, m represents the position of the signal, and M is the size of the blur kernel. Based on (1), the clear signal x can be obtained by

$$x[n] = \frac{y[n]}{k[0]} - \frac{\sum_{m=1}^{M} k[m]x[n-m]}{k[0]}, \tag{2}$$

which is an M-th order infinite impulse response (IIR) model. By expanding the second term of (2), we find that the deconvolution process requires an infinite signal information as follows:

$$\begin{aligned} x[n] &= \frac{y[n]}{k[0]} - \sum_{m=1}^{M} \frac{k[m]}{k[0]}\left(\frac{y[n-m]}{k[0]} - \frac{\sum_{l=1}^{M} k[l]x[n-m-l]}{k[0]}\right) \\ &= \frac{y[n]}{k[0]} - \sum_{m=1}^{M} \frac{k[m]y[n-m]}{k[0]^2} + \sum_{m=1,l=1}^{M,M} \frac{k[m]k[l]x[n-m-l]}{k[0]^2} \\ &= ..., \end{aligned} \tag{3}$$

In fact, if we assume that the boundary of the image is zero, (3) is equivalent to applying an inverse filter to y. As shown in Figure 2, the non-zero region of the inverse filter is much larger than the blur kernel, which means that a large receptive field should be considered in the deconvolution.

Thus, if we use a CNN to approximate (3) (which means that the CNN actually learns the weights of y in (3)), where the basic operations of CNN are convolution and non-linear activation, a large receptive field should be considered to cover the positions that are used in (3). As such, conventional CNN-based methods [21, 22] usually need to have a large network structure to achieve this goal. However, this inevitably leads to large model size, which is computationally expensive.

From (2), we find that only a few coefficients, which is $k[m], m = 0, 1, ..., M$, are needed in the IIR deblurring model. This means that a few parameters are needed to deblur an image as long as we can find an appropriate operation to cover a large enough receptive field. Thus, if we develop a network based on (2), the model size will be much smaller.

We note that the spatially variant RNN [19] satisfies the above requirements. However, directly using the RNN connection strategy [19] cannot achieve our goals, as it does not fuse the information from different filtering directions between consecutive RNNs and each output pixel of the RNN will only consider information from the column and row that it is in.

To consider 2D information with a large receptive field in our network, we insert a convolution layer between consecutive RNNs. Figure 3 shows a toy example of fusing the information of the spatial RNN from different directions by a CNN. It shows that by adding a CNN after the RNN,

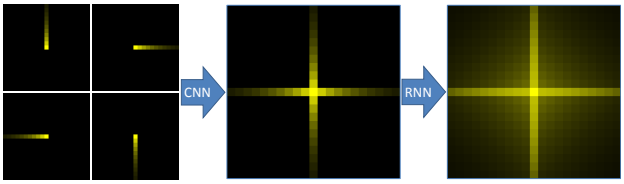


Figure 3. A toy example of fusing the information of the spatial RNN from different directions by the CNN. (a) shows the four receptive fields of the spatial RNN from four different directions, and each RNN only considers 1D information. Without adding the CNN between RNNs, the upper left part of the first RNN in (a) will connect to the top left corner part of the second RNN according to the corresponding directions. Thus, the receptive fields after two consecutive RNNs are still the same as Figure 3(a), which cannot be considered as 2D information. (b) is the receptive field by adding a 1×1 CNN after the RNN to fuse the information from the RNN. (c) is the receptive field by adding another RNN and the output now can consider 2D information of a large receptive field. The non-black region is the receptive field of the center pixel.

information from different directions can be fused and the final receptive field can cover a large 2D region after another RNN. In this way, the spatial RNN can be used to cover a large 2D region with a small number of parameters. The other advantage of the spatially variant RNN is that its weights can be learned from another network. It is similar to the traditional deblurring method, which estimates a blur kernel first and uses this kernel to recover the clean image. As a result, the network does not need to remove different blurs with the same weights, which will enlarge the model size. In addition, different weights can be learned for different locations, which is suitable for spatially variant blur in dynamic scenes.

3.2. Network Structure

We propose a novel spatially variant RNN to solve the dynamic scene deblurring problem. We first use a feature extraction network to extract features from the blurry images. The spatially variant RNN is then used for deblurring in the feature space according to the RNN weights, which are learned from a weight generation network. We add a convolution layer after every RNN to fuse the information from different directions. Finally, we use an image reconstruction network to reconstruct the clean image.

Figure 4 shows the proposed network architecture. Table 1 summarizes the network configurations and contains four parts: feature extraction, RNN weight generation, RNN deconvolution (including convolution layer after every RNN) and image reconstruction. There are two convolution layers in the feature extraction part. The feature maps are downsampled by half to reduce the memory cost of the network. The four RNNs are then used to filter these features. Every RNN has four directions. We use a convolution layer to fuse the information from the RNN output. To compute the pixel-wise weights of the RNN, we use a 14 layers CNN (i.e., conv3-conv16 in Figure 4). We fine-tune the weights of conv3-conv11 from the first nine layers of VGG16 [30] in order to have a good initialization. The image reconstruction

(a) clean image (b) blur kernel (c) blurry image (d) inverse filter (e) deblurred image

Figure 2. The deconvolution process needs large image regions. (a) is a clean image. (c) is obtained by blurring (a) with the motion kernel from [16] as shown in (b). (d) is a regularized inverse filter from Wiener filtering [37], which can remove the motion blur. (e) is the deblurred image. The non-zero region of the inverse filter is much larger than the blur kernel.

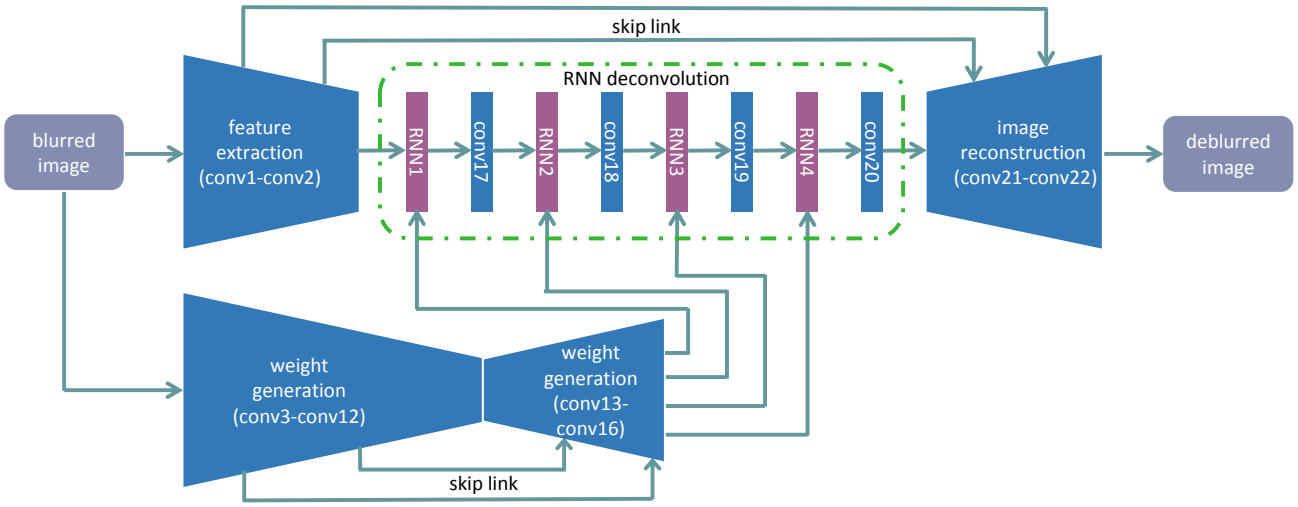


Figure 4. The proposed network structure. Two CNNs are used to extract features and generate pixel-wise weights for the spatially variant RNN. For RNN deconvolution, four RNNs are applied to the feature maps to remove blur and every RNN considers four directions. A convolution layer is added after every RNN to fuse the information. Four skip links are added between feature extraction and image reconstruction as well as in weight generation. One CNN is used in image reconstruction to estimate the final deblurred image. The non-linear function ReLU or Leaky ReLU is used in each CNN. See Table 1 for detailed CNN configurations.

Table 1. Configurations of the network. The feature maps are downsampled by convolution with stride 2 and upsampled by bilinear interpolation. Four skip links are added and we concatenate on conv1 with resize1, conv2 with conv20, conv8 with resize2, as well as conv6 with resize3.

	feature extraction		RNN deconvlution								image reconstruction		
layer	conv1	conv2	rnn1	conv17	rnn1	conv18	rnn3	conv19	rnn4	conv20	conv21	resize1	conv22
size	11	7		3		3		3		3	9		5
channel	16	32	32	32	32	32	32	32	32	32	16		3
stride	1	↓2		1		1		1		1	1	↑2	1
concatenate										conv2		conv1	

	RNN weights generation																		
layer	conv3	conv4	pool1	conv5	conv6	pool2	conv7	conv8	conv9	pool3	conv10	conv11	conv12	resize2	conv13	conv14	resize3	conv15	conv16
size	3	3		3	3		3	3	3		3	3	3		3	3		3	3
channel	64	64		128	128		256	256	256		512	512	256		128	128		256	512
stride	1	1	↓2	1	1	↓2	1	1	1	↓2	1	1	1	↑2	1	1	↑2	1	1
concatenate														conv8			conv6		

part can estimate the deblurred image from the RNN filtered feature maps. To avoid gradient vanishing and to accelerate training, four skip links are added by concatenating their inputs. We use bilinear interpolation, instead of a deconvolution layer, to upsample the feature maps and avoid grid artifacts generated by the deconvolution layer. Rectified Linear Unit (ReLU) is added after every convolution layer of the weight generation network, except for the last convolution layer after which a hyperbolic tangent (tanh) layer is added to constrain the RNN weights to be between 0 to 1, just as in [19]. Leaky ReLU with negative slope 0.1 is also added after every convolution layer in the feature extraction network, RNN fusion and image reconstruction network, except for the last convolution layer in the whole network.

3.3. Network Training

The proposed model is trained on the training set for dynamic scene deblurring [21] as well as deep video deblurring [32]. As the blur in [32] is very small for most of the images, 50% of the images add motion blur with maximum 20 pixels blur and 50% of the images add foreground objects, which is from Caltech 101 [17], with maximum 20 pixels blur. We

augment the training data by random cropping, resizing, rotation and color permutation. The patch size is 128 and every batch contains 20 patches. We implement the proposed algorithm using Caffe [12]. The L_2 loss is used to train the network. The spatially variant RNN is implemented by the approach [19]. CNN weights are initialized by the Xavier method, except for conv3-conv11 in the weight generation network, which are fine-tuned from the first nine layers of VGG16 [30]. Adam is used to optimize the network. The learning rate, momentum, momentum2 and weight decay are 0.0001, 0.9, 0.999 and 0.000001, respectively. According to our experiments, the network converges after 200,000 iterations.

4. Experimental Results

We evaluate our method on the dynamic scene deblurring dataset [21] and compare it with state-of-the-art image deblurring algorithms, including conventional uniform deblurring [41, 24], non-uniform deblurring [36], and CNN based dynamic scene deblurring [33, 7, 21] in terms of PSNR and SSIM. We have retrained the network by Liu et al. [19] using the same dataset of our network for fair comparison though it is not designed for image deblurring. In addition, we compare the visual results of the proposed algorithm with those of the other algorithms on the real blurry dataset [3]. The trained models, source code, and datasets are publicly available on the authors' websites. Due to the page limit, we only show a small portion of the results. More results are included in the supplemental material.

4.1. Quantitative Evaluations

Table 2 shows the average PSNR and SSIM values of the restored images on the test datasets [21]. The proposed method performs favorably against with state-of-the-art algorithms in terms of PSNR and SSIM. The generated results have much higher PSNR and SSIM values.

Figure 5 shows several examples from the test set. Due to the moving objects (e.g., cars) and camera shake, the blurry images contain significant blur effect. The conventional non-uniform deblurring methods [36, 41, 33, 24] are not able to generate clear results as these methods focus on the blur caused by camera shake. The CNN-based methods [21, 33, 7] are designed for dynamic scene deblurring. However, these methods are not able to remove large blur due to the limited receptive field in their networks. We note that Liu et al. [19] develop a hybrid network including a CNN and RNN for image processing. However, this method is less effective for image deblurring as shown in Figure 5(f). In contrast, the proposed algorithm recovers the clear images with finer details and clearer structures.

4.2. Qualitative Evaluations

We further qualitatively evaluate the proposed method on the real blurry images from [3]. Figure 6 shows several real images and the results generated by the proposed method and state-of-the-art methods. The conventional deblurring methods [36, 41, 24] fail to generate clear images. We note that Sun et al. [33] develop a CNN-based method for motion blur kernel estimation. However, the final recovered images contain some artifacts due to imperfect estimated blur kernels. Compared to the CNN-based methods [21], the proposed method generates much clearer images with clearer structures and characters. More experimental results are included in the supplemental material.

4.3. Run-Time and Model Size

We evaluate our method and state-of-the-art methods on the same PC with an Intel(R) Xeon(R) CPU and a Nvidia Tesla K80 GPU. As shown in Table 3, the conventional non-uniform deburring methods have high computational cost as these methods usually need to solve highly non-convex optimization problems. Although Sun et al. [33] and Gong et al. [7] develop CNN algorithms to estimate motion blur, both of them need a conventional non-blind deblurring algorithm to generate the final clean image, which increases the computational cost. The method in [21] uses a multi-scale CNN to increase the receptive field to estimate clear images and spends much less computational time compared with the conventional algorithms. However, a multi-scale scheme inevitably increases the computational load and it is still not efficient compared to the proposed method. Furthermore, the model size of [21] is much larger than the proposed method as shown in Table 3. As the proposed method includes a novel spatially variant RNN with fewer parameters according to the analysis in Section 3.1, the model size of the proposed method (37.1MB) is much smaller than that of [21] (303.6MB) (Table 3). In addition, the running time of proposed method is 10.0x faster than [21].

5. Analysis and Discussions

In this section, we discuss the effect of the proposed method and clarify the relationship between the proposed method with other deep learning-based methods.

5.1. Effectiveness of the Spatially Variant RNN

To demonstrate the effectiveness of the spatially variant RNN, we remove the RNNs from the network and keep the weights of the rest network. As can be seen in Figure 7(b), the deblurred result without using RNNs still contains a significant blur residual. By adding the spatially variant RNNs, a clean image can be recovered as shown in Figure 7(c). This shows that it is the RNNs, rather than other parts, that remove the blur in the proposed network.

Part of the RNN weights of Figure 7(a) are shown in Figure 8(b) to (e). In order to roughly show the motion of blur, we use FlowNet 2.0 [11] to estimate the optical flow as shown in Figure 8(a). According to the optical flow results, part of the foreground people move differently relative to the rest of the image. At the same time, these foreground people regions also have different RNN weights, which demonstrates that the weight generation network can detect different blur and the RNN weights act as the estimated blur kernel to recover the clean image.

Table 2. Quantitative evaluation on the dynamic scene deblurring dataset [21], in terms of PSNR and SSIM.

method	Whyte [36]	Xu [41]	Sun [33]	Pan [24]	Liu [19]	Nah [21]	Gong [7]	proposed
PSNR	24.5312	20.2976	25.3098	23.5049	25.7464	28.4898	26.0576	**29.1872**
SSIM	0.8458	0.7407	0.8511	0.8336	0.8654	0.9165	0.8632	**0.9306**

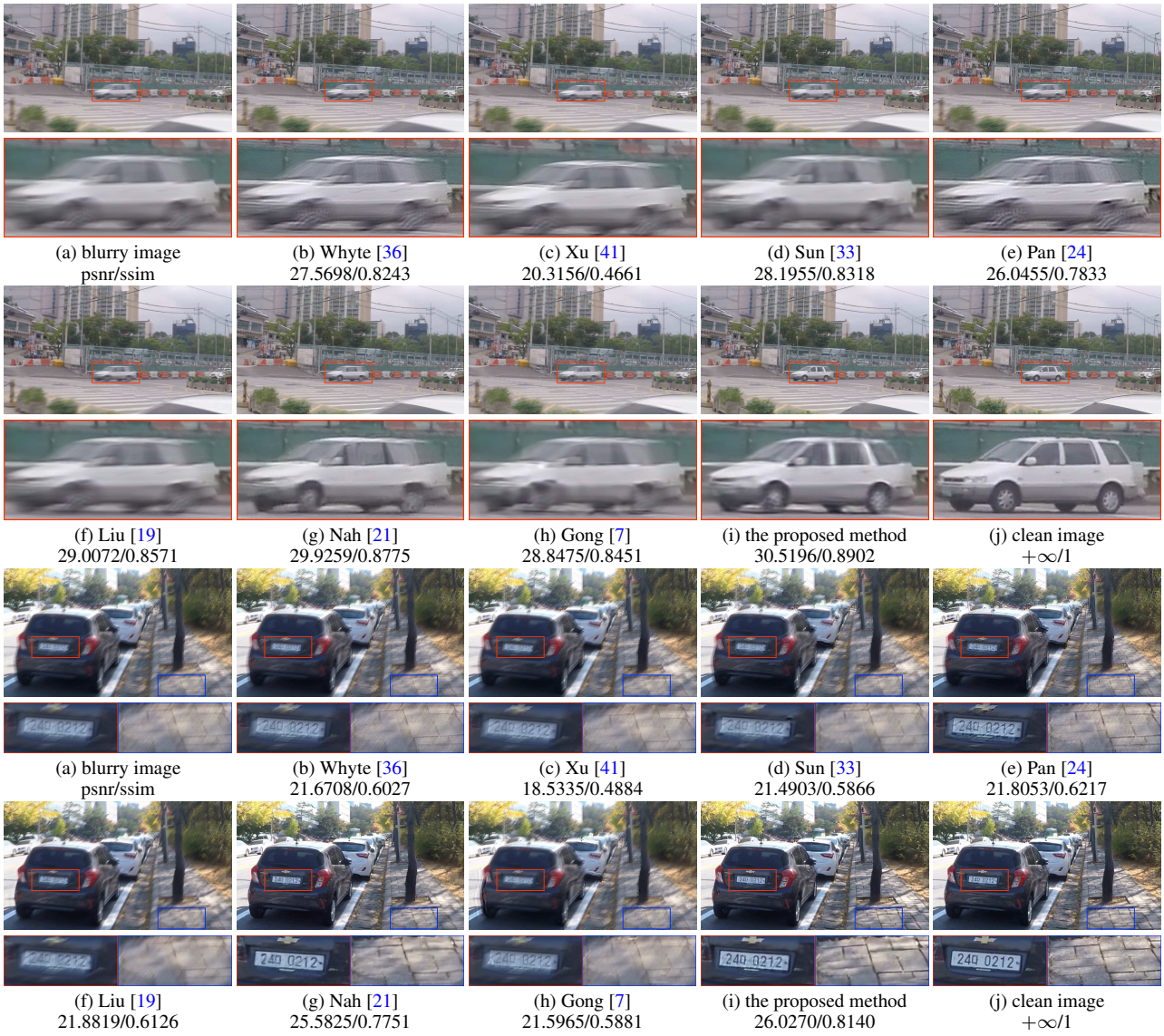


Figure 5. Quantitative evaluations on the dynamic scene deblurring dataset [21]. The proposed method generates much clearer images with higher PSNR and SSIM values.

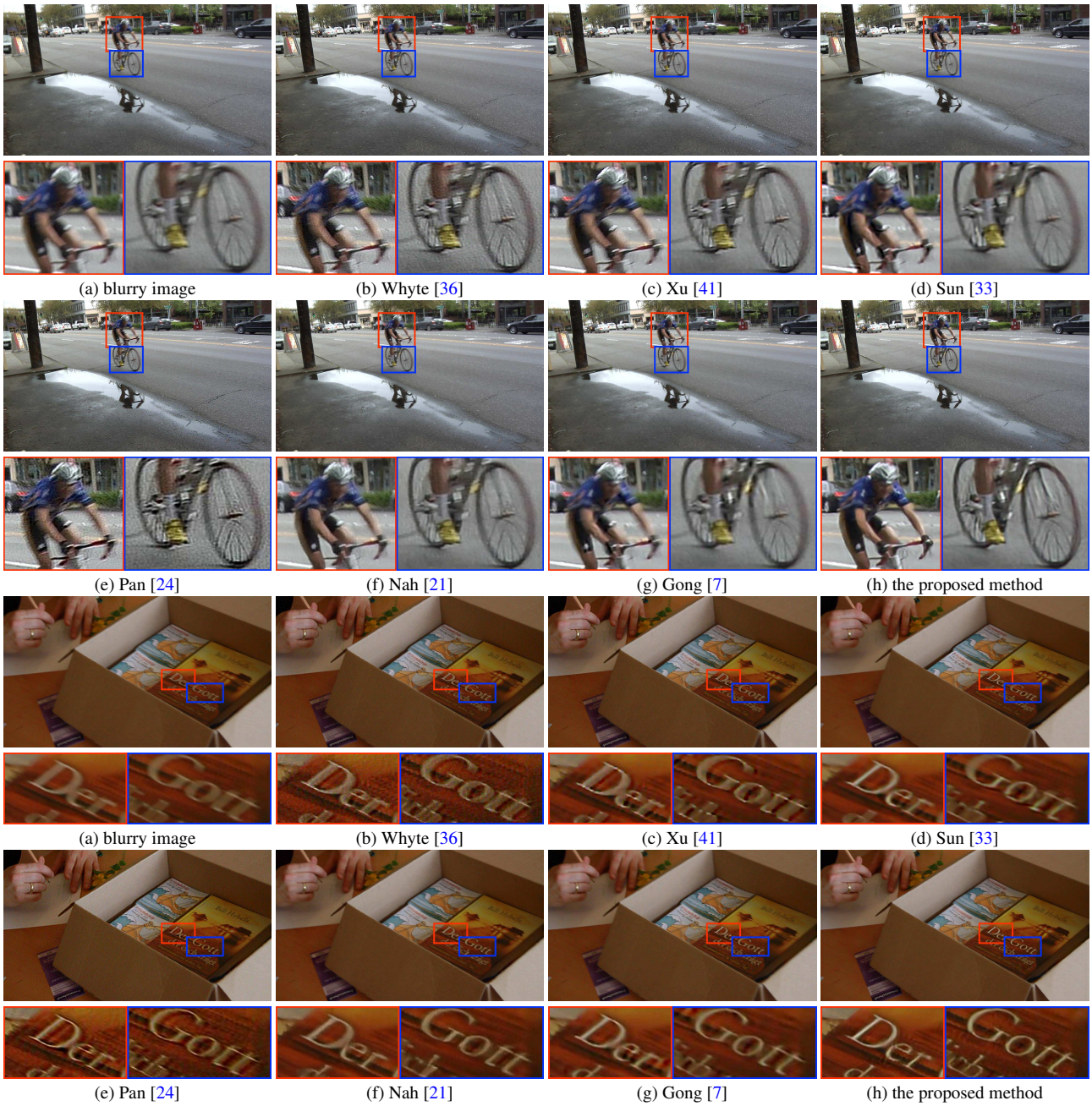
Table 3. Running time and network model size for an image with the size of 720×1280 pixels. All existing methods use their publicly available scripts. A "-" indicates that the result is not available.

	Whyte [36]	Xu [41]	Sun [33]	Pan [24]	Nah [21]	Gong [7]	proposed
time(sec)	700	3800	1500	2500	15	1500	**1.4**
size(MB)	–	–	54.1	–	303.6	41.2	**37.1**

5.2. Relation with Deep Learning-based Methods

According to [40, 38], a large region should be considered for deblurring in CNN-based methods even though the blur kernel is small. To solve dynamic scene deblurring, Nah et al. [21, 22] use a multi-scale scheme and deep network structure to cover a large receptive field. In addition, the sizes of their networks are too large as the network should handle different blurs with the same weights.

We note that Liu et al. [19] propose a hybrid neural network for image filtering and inpainting. They simply connect the 1D RNNs, which are from four directions. As a result, the network only fuses the information from a single column



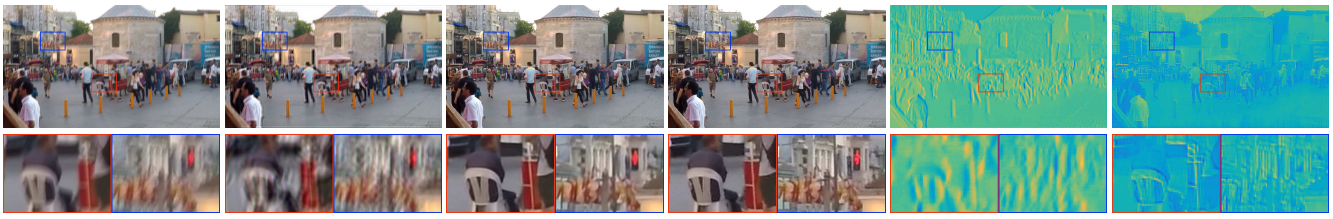
(a) blurry image (b) Whyte [36] (c) Xu [41] (d) Sun [33]

(e) Pan [24] (f) Nah [21] (g) Gong [7] (h) the proposed method

(a) blurry image (b) Whyte [36] (c) Xu [41] (d) Sun [33]

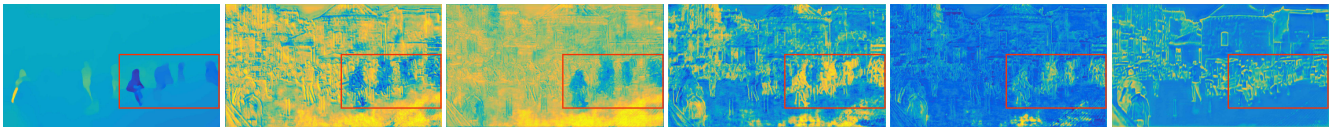
(e) Pan [24] (f) Nah [21] (g) Gong [7] (h) the proposed method

Figure 6. Qualitative evaluations on the real blurry dataset [32]. The proposed method generates much clearer images with clearer structures and characters.



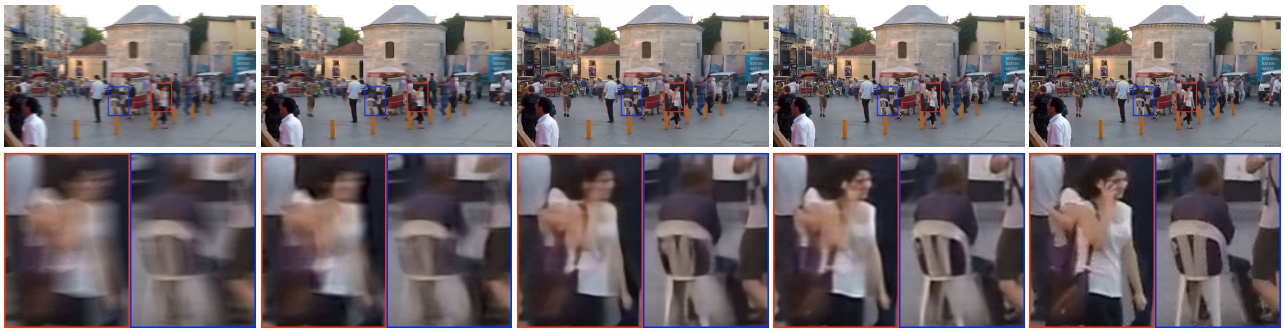
(a) blurry image (b) without RNNs (c) with RNNs (d) clean image (e) before RNNs (f) after RNNs

Figure 7. The effectiveness of the proposed RNNs. (e) and (f) are some selected feature maps before and after the RNNs. The RNNs are able to help remove the blur.



(a) (b) (c) (d) (e) (f)

Figure 8. Visualizations of the learned RNN weights. (a) is the optical flow from the adjacent frames of Figure 7(a) according to FlowNet 2.0 [11]. (f) is the selected RNN weights of the spatially variant RNN from [19]. (b)-(e) are selected RNN weights of the spatially variant RNN from the proposed method. According to (a), some of the foreground objects (e.g., people) have different motions compared to other parts. The generated RNN weights are able to distinguish the different moving objects in (b) - (e). This demonstrates that the proposed weight generation network can detect different blurs. However, the method by [19] only extracts the edges, which do not reflect the motion of the objects.



(a) blurry image PSNR/SSIM | (b) without RNNs 25.8015/0.8019 | (c) weights generation 28.0155/0.8746 | (d) proposed 29.8986/0.8995 | (e) clean $+\infty$/1

Figure 9. Visual results for the ablation study on the dynamic scene dataset [21]. The proposed method generates clearer images with higher PSNR and SSIM values relative to the networks with only CNNs. Refer to the text for details.

Table 4. An ablation study on the dynamic scene dataset [21] in terms of PSNR and SSIM. The proposed network is compared with the network without skip links, the network without CNN between RNNs, the network without RNNs as well as only using the weight generation network structure to deblur. Refer to the text for details.

network	w/o skip	w/o convs	w/o RNNs	weights generation	proposed
PSNR	25.7687	27.8365	26.0413	27.6835	**29.1872**
SSIM	0.9002	0.9087	0.8689	0.9120	**0.9306**

and row that contains the output pixels, instead of considering the information from the whole image. This leads to a limited receptive field. Thus, [19] cannot be directly applied to image deblurring as the problem is quite different from the filtering problem and needs a large receptive field. As shown in Figure 8(f), the method by [19] does not estimate reliable RNN weights compared to the proposed algorithm. The final deblurred results by [19] are still blurry as shown in Figure 5(f).

In contrast, we propose a 3×3 convolution layer between each consecutive RNN to let the proposed network consider the 2D information of the image. Thus, a much larger receptive field can be involved (Figure 2(c)). In addition, we propose an auto-encoder scheme to further reduce the model size and save memory cost of the proposed network. The proposed method generates reliable feature maps (Figure 8(b)-(e)) and much clearer images (Figure 5(i)).

5.3. Ablation Study

The proposed network contains four parts: feature extraction network, weight generation network, RNNs (including convolution layers between RNNs) and image reconstruction network. Here, we compare the proposed network with the network without RNNs (but keeping the convolution layers between the RNNs), and with only the weight generation network (using it to deblur directly). We also compare the proposed network with the network without skip links as well as the network without the convolution layers between RNNs. We train these four networks using the same training strategy as in Section 3.3. As shown in Table 4 and Figure 9, the proposed network cannot work well if any part is removed.

6. Conclusions

In this paper, we propose a novel end-to-end spatially variant recurrent neural networks (RNNs) for dynamic scene deblurring, where the weights of the RNNs are learned by a deep CNN. We analyze the relationship between the proposed spatially variant RNN and the deconvolution process, and show that the spatially variant RNN is able to model the deblurring process. With the proposed RNNs, the trained model is significantly smaller and faster in comparison with existing CNN-based deblurring methods. Both quantitative and qualitative evaluations on the benchmark datasets demonstrate the effectiveness of the proposed method in terms of accuracy, speed, and model size.

Acknowledgements. This work have been supported in part by the national key research and development program (No. 2016YFB1001001), NSFC (No. 61522203, 61732007 and 61772275), NSF CAREER (No. 1149783), the National Ten Thousand Talent Program of China (Young Top-Notch Talent), and gifts from Adobe, Toyota, Panasonic, Samsung, NEC, Verisk, and Nvidia.

References

[1] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, 2012. 2

[2] A. Chakrabarti. A neural approach to blind motion deblurring. In *ECCV*, 2016. 2

[3] S. Cho, J. Wang, and S. Lee. Video deblurring for hand-held cameras using patch-based synthesis. *TOG*, 2012. 5

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 2

[5] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, 2013. 2

[6] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017. 2

[7] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. v. d. Hengel, and Q. Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *CVPR*, 2017. 1, 2, 5, 6, 7

[8] M. Hradiš, J. Kotera, P. Zemcík, and F. Šroubek. Convolutional neural networks for direct text deblurring. In *BMVC*, 2015. 2

[9] T. Hyun Kim, B. Ahn, and K. Mu Lee. Dynamic scene deblurring. In *ICCV*, 2013. 1, 2

[10] T. Hyun Kim and K. Mu Lee. Segmentation-free dynamic scene deblurring. In *CVPR*, 2014. 1, 2

[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 5, 8

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[13] J. Kim, J. Lee, and K. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 2

[14] J. Kim, J. Lee, and K. Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 2

[15] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017. 2

[16] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009. 4

[17] F.-F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU*, 2007. 4

[18] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep joint image filtering. In *ECCV*, 2016. 2

[19] S. Liu, J. Pan, and M.-H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, 2016. 2, 3, 4, 5, 6, 8

[20] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep fully convolutional encoder-decoder networks with symmetric skip connections. In *NIPS*, 2016. 2

[21] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[22] M. Noroozi, P. Chandramouli, and P. Favaro. Motion deblurring in the wild. In *German Conference on Pattern Recognition*, 2017. 1, 2, 3, 6

[23] J. Pan, Z. Hu, Z. Su, H.-Y. Lee, and M.-H. Yang. Soft-segmentation guided object motion deblurring. In *CVPR*, 2016. 1, 2

[24] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *CVPR*, 2016. 5, 6, 7

[25] J. G. Proakis and D. K. Manolakis. *Digital signal processing, principles, algorithms, and applications*. Pentice Hall, 1996. 2

[26] J. S. Ren, L. Xu, Q. Yan, and W. Sun. Shepard convolutional neural networks. In *NIPS*, 2015. 2

[27] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, 2016. 2

[28] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, 2013. 2

[29] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *TPAMI*, 2016. 2

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 5

[31] Y. Song, J. Zhang, S. He, L. Bao, and Q. Yang. Learning to hallucinate face images via component generation and enhancement. In *IJCAI*, 2017. 2

[32] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring. In *CVPR*, 2017. 4, 7

[33] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, 2015. 1, 2, 5, 6, 7

[34] P. Svoboda, M. Hradiš, L. Maršík, and P. Zemcík. Cnn for license plate motion deblurring. In *ICIP*, 2016. 2

[35] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *ICCV*, 2015. 2

[36] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *IJCV*, 2012. 5, 6, 7

[37] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*. MIT press Cambridge, MA, 1949. 4

[38] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014. 2, 6

[39] L. Xu, J. S. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *ICML*, 2015. 2

[40] L. Xu, X. Tao, and J. Jia. Inverse kernels for fast spatial deconvolution. In *ECCV*, 2014. 6

[41] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013. 5, 6, 7

[42] X. Xu, J. Pan, Y. Zhang, and M.-H. Yang. Motion blur kernel estimation via deep learning. *TIP*, 2018. 2

[43] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang. Learning to super-resolve blurry face and text images. In *ICCV*, 2017. 2

[44] J. Zhang, J. Pan, W.-S. Lai, R. Lau, and M.-H. Yang. Learning fully convolutional networks for iterative non-blind deconvolution. In *CVPR*, 2017. 2

[45] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 2

[46] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 2