

Dynamic Scene Deblurring with Parameter Selective Sharing and Nested Skip Connections

Hongyun Gao^{1,2} Xin Tao² Xiaoyong Shen² Jiaya Jia^{1,2}

¹ The Chinese University of Hong Kong ² YouTu Lab, Tencent

{hygao, leojia}@cse.cuhk.edu.hk {xintao, dylanshen}@tencent.com

Abstract

Dynamic Scene deblurring is a challenging low-level vision task where spatially variant blur is caused by many factors, e.g., camera shake and object motion. Recent study has made significant progress. Compared with the parameter independence scheme [19] and parameter sharing scheme [33], we develop the general principle for constraining the deblurring network structure by proposing the generic and effective selective sharing scheme. Inside the subnetwork of each scale, we propose a nested skip connection structure for the nonlinear transformation modules to replace stacked convolution layers or residual blocks. Besides, we build a new large dataset of blurred/sharp image pairs towards better restoration quality. Comprehensive experimental results show that our parameter selective sharing scheme, nested skip connection structure, and the new dataset are all significant to set a new state-of-the-art in dynamic scene deblurring.

1. Introduction

Image blur, caused by camera shake, object motion or out-of-focus, is one of the most common visual artifacts when taking photos. Image deblurring, *i.e.*, restoring the sharp image from the blurred one, has been an important research area since decades ago. Due to the ill-posed nature, particular assumptions are required to model different types of uniform, non-uniform, and depth-aware blur. Many natural image priors [1, 27, 2, 35, 36, 21, 22] were proposed to regularize the solution space and advance the deblurring research.

Compared with the blur caused by only camera translation or rotation, dynamic blur is more realistic and also very challenging since spatially variant blur is the combined effect of multiple factors. Previous dynamic scene deblurring methods [12, 13, 20] usually rely on an accurate image segmentation mask to estimate different blur kernels for corresponding image regions, and employ complex op-

timization methods to restore the latent image. Recently, learning-based methods were proposed to facilitate the deblurring process by either replacing some steps in the traditional framework [31, 26, 5] or learning the end-to-end mapping from blurred to latent images [19, 33, 37].

Parameter Selective Sharing Nah *et al.* [19] first proposed the “coarse-to-fine” deblurring neural network by progressively restoring the sharp image in a coarse-to-fine manner. This approach built a deep neural network with independent parameters for each scale. It does not consider the parameter relation across scales. Tao *et al.* [33] advanced a scale-recurrent network to perform deblurring in different scales by shared parameters. Albeit concise and compact, this parameter sharing scheme neglects scale-variant property of features, which are crucial for respective restoration in each scale.

We believe scale-recurrent structure belongs to a broader set of parameter selective sharing strategies. The rationale behind it is that image blur is highly dependent on scale – large blur in a fine scale could be invisible when the image is downsampled to coarse scales. Particular modules perform scale-variant operations and thus cannot be shared, while others perform similar scale-invariant transform that benefit from shared parameters. We delve deep into this selective sharing strategy, and elaborate on why specific modules can or cannot be shared. We also discuss various sharing strategies across and within scales. These findings lead to the general principle of selective parameter sharing that can benefit deblurring system design in future.

Nested Skip Connections Skip connection was widely used in recent CNNs. In ResNet [7] and ResBlocks [19, 17], the short-term skip connection that adds the input to the output after two or more convolution layers, is the key factor to produce superior results in object detection, deblurring and super resolution. With this short-term skip connection, the gradient-vanishing issue can be largely overcome; very deep networks can thus be built and optimized easily. The long-term skip connection, common in encoder-decoder networks, links feature maps from bottom layers to

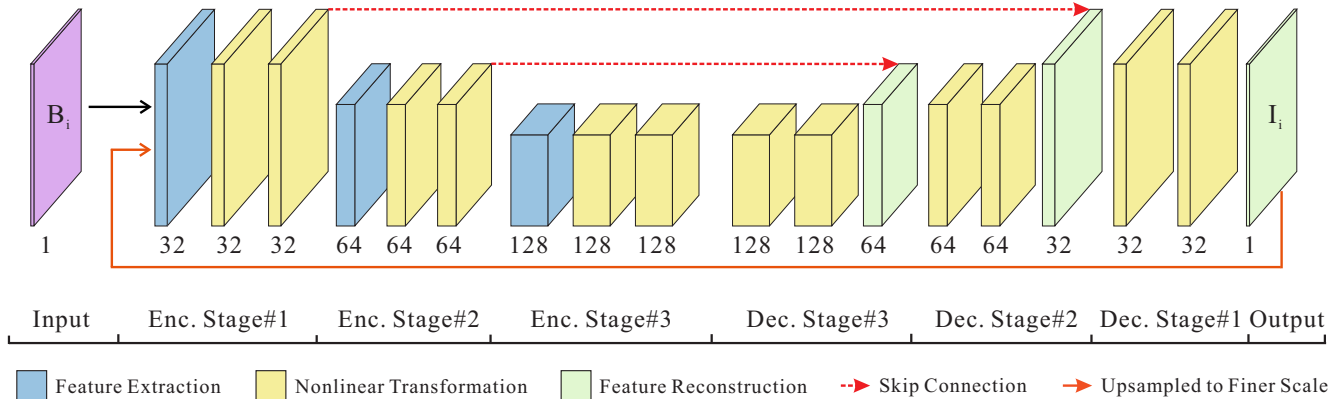


Figure 1. Our encoder-decoder subnetwork includes 3 encoder stages and 3 decoder stages. It has 3 kinds of modules, *i.e.*, feature extraction, nonlinear transformation and feature reconstruction.

top ones. This scheme allows the information to be back-propagated more flexibly and pass image details from bottom layers to top ones for better detail reconstruction.

However, we note that both short- and long-term skip connections do not have intersecting paths since they only consider the first-order residual learning according to our analysis in the following subsection. We instead propose a nested skip connection structure that corresponds to higher-order residual learning for the transformation modules in our deblurring network.

Dataset For training deep deblurring networks, sufficient paired blurred/sharp images are essential. Although the GoPro dataset [19] provides 3,214 pairs, there exist flaws in a portion of images that may adversely affect network training. We thus build a larger and higher-quality dataset towards training better deblurring networks by overcoming the flaws. It has 5,290 blurred/sharp image pairs following the procedures of [19]. We compare the same network trained with only GoPro dataset and that trained with GoPro and our dataset together. Experimental results demonstrate that our dataset is favorably helpful both quantitatively and qualitatively.

The main contributions of our work are as follows.

- We analyze the parameter strategies for the deblurring networks and propose a generic principled parameter selective sharing scheme with both independent and shared modules for the subnetworks in each scale.
- We propose a nested skip connection structure for the feature transformation modules in the network, which corresponds to higher-order residual learning in individual transformation modules.
- We establish a larger and higher-quality dataset with 5,290 blurred/sharp image pairs to help network training. It is publicly available to advance general image deblurring research.

2. Related Work

In this section, we briefly review dynamic scene deblurring methods, CNN parameter sharing schemes and skip connection used in network structures.

Dynamic Scene Deblurring After the work of Kim *et al.* [12], dynamic scene blurring became a tractable topic for scenes that are not static and the blur is caused by camera shake and complex object motion. However, the performance of this method highly relies on the accuracy of motion segmentation. Later, Kim and Lee [13] assumed that motion is locally linearly varying, and hence proposed a segmentation-free approach to handle this problem. In [20], a segmentation confidence map was used to reduce segmentation ambiguity between different motion regions.

Recently, several methods [31, 5, 19, 33, 37] used deep learning to better solve the task in terms of restoration quality and adaptiveness to different situations. Sun *et al.* [31] employed a classification CNN to predict blur direction and strength of a local patch. A dense motion field is obtained via Markov Random Fields (MRF) from the sparse blur kernel. The final latent image is generated by the non-blind deblurring method [39]. Gong *et al.* [5] utilized a fully convolutional network to estimate the dense heterogeneous motion flow from the blurred image and still used the method of [39] to recover the latent image. Motivated by the traditional “coarse-to-fine” optimization framework, Nah *et al.* [19] proposed a multi-scale deblurring CNN to progressively restore sharp images in multiple scales in an end-to-end manner. Tao *et al.* [33] improved the pipeline to model the scale-recurrent structure with shared parameters. Zhang *et al.* [37] proposed a RNN to model the spatially varying blur where the pixel-wise weights of the RNN are learned from a CNN.

CNN Parameter Sharing Despite widely adopted in temporal and sequential data processing, parameter sharing

is still a new try for image algorithms. In fact, CNN parameter sharing incorporates large context information but at the same time maintains the model size. It is effective in tasks of *e.g.*, object classification [28], scene parsing [23], object recognition [16], image super resolution [11] and dynamic scene deblurring [33]. Specifically, Socher *et al.* [28] used a CNN to first learn translational invariance features and applied the same network recursively to learn hierarchical feature representations in a tree structure. Pinheiro and Collobert [23] proposed a recurrent structure composed of two or three identical CNNs with shared parameters. Liang and Hu [16] incorporated recurrent connections into each convolutional layer to integrate different levels of context information. Kim *et al.* [11] utilized a deep recursive layer in the image super-resolution network to increase receptive fields. Tao *et al.* [33] progressively restored the latent image from coarse to fine scales using a scale-recurrent network.

Skip Connections As the neural networks become deeper, the gradient-vanishing issue severely hampers effective training. Many architectures were proposed to address this issue. Highway network [29] was among the first to train very deep networks using bypassing paths. ResNet [7] used identity mapping to skip one or more layers and enabled training substantially deeper networks. DenseNet[8] further connected each layer to every other within a dense module to propagate all preceding information for succeeding processing.

Despite the success in high-level vision tasks, skip connections were also widely used in image processing. Input images are often added to the reconstructed ones in image/video restoration [10, 30, 32, 9], since learning the residual image through a CNN is much easier than reconstructing decent output. Further, skip connections were also used between the internal layers [34, 6, 38] to fully utilize different levels of features. After the seminal work of U-net [25], skip connections between the corresponding encoder and decoder stages were widely used as an effective architecture for pixel-wise regression in optical flow estimation [3], image restoration [18] and raindrop removal [24].

3. Proposed Method

As illustrated in Fig. 1, our network is composed of several stacked encoder-decoder subnetworks, from which sharp images at different scales are produced and are fed into the subnetwork in the next scale as input. Different from stacked ResBlocks in [33], our network consists of 3 kinds of modules to perform different functions, *i.e.*, feature extraction, nonlinear transformation and feature reconstruction. Compared with [33], we make better use of parameters and design a new nested skip connection structure for the nonlinear transformation modules.

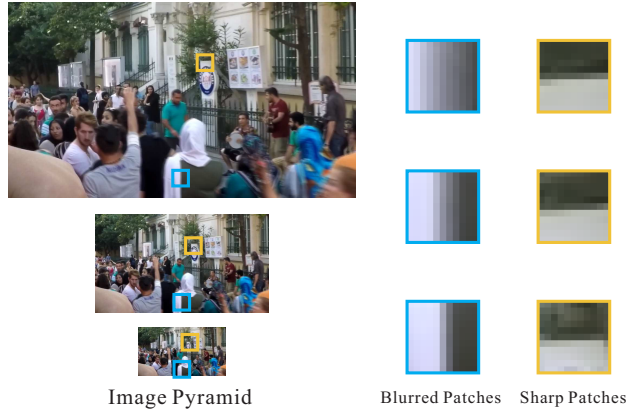


Figure 2. An example of scale-variant and scale-invariant features.

3.1. Parameter Selective Sharing

Although both methods of [19, 33] progressively restore the sharp images in a coarse-to-fine manner, they utilize different parameter strategies to achieve the objective. The parameter independence scheme in [19] assigns independent parameters for each scale. It, however, lacks constraints to handle different scales. The parameter sharing scheme in [33] constrains the solution space using shared parameters in different scales. We consider two aspects regarding the parameter issue. The first is on what kind of parameters can be shared across scales. The second issue is whether the parameters of different modules within one scale can be shared or not.

Parameter Independence Fig. 2 shows a typical blurred image in dynamic scenes. The background building is roughly clear but the foreground people are blurred. When we employ the “coarse-to-fine” framework to perform deblurring, different features should be handled. Here, we analyze two typical regions in the image pyramid by cropping 11×11 patches at the same location. One is a sharp region in the background building, and the other is a blurred region in the foreground people.

The features in the sharp region are similar, since the downsampled sharp edges are still sharp. However, the features in the blurred region are different, since a blurred edge becomes sharp after scaling down. If the feature extraction module is shared across scales, it cannot simultaneously extract sharp and blurred features. When learned from sharp features in the coarse scale, the shared feature extraction module cannot extract blurred features in the fine scale.

With this observation, we relax the parameter sharing scheme [33] and assign independent parameters for the feature extraction module in each stage of the subnetwork, such that the network can automatically extract the most discriminative features in each scale. As shown in Fig. 3, with independent feature extraction modules, our param-

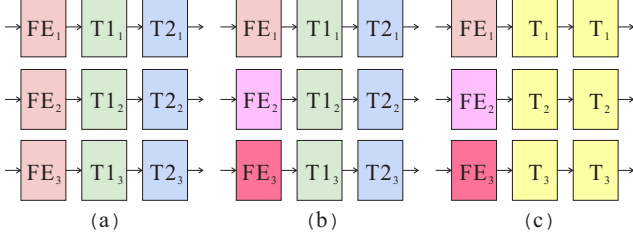


Figure 3. **Parameter sharing strategies in the encoder stage of our subnetwork.** The blocks in three rows indicate “coarse-to-fine” strategy from coarse to fine scales. “FE” is the feature extraction module. “T” is the nonlinear transformation module. The modules in the same color share the same parameters. (a) Scale-recurrent structure with the same parameters across scales. (b) Modified scale-recurrent structure with independent feature extraction modules. (c) Modified version of (b) with shared nonlinear transformation parameters within a stage and also across scales.

ter scheme (b) is different from scale-recurrent modules in (a). After the features are extracted and transformed in the encoder part, the feature reconstruction modules gradually reconstruct the features back to the sharp image. Since scale-variant features are extracted using independent parameters, the corresponding feature reconstruction modules are also with independent parameters to process the scale-variant features.

Parameter Sharing After extracting scale-variant features, we transform them to the corresponding sharp features. The nonlinear transformation modules across different scales perform the same blur-to-sharp transformation. Thus parameters can be shared across scales, which is confirmed in the scale-recurrent structure [33]. This **inter-scale** parameter sharing scheme is shown in Fig. 3(b).

Motivated by the traditional iterative image deblurring, which uses the same solver iteratively, we hypothesize there also exists **intra-scale** parameter sharing between the nonlinear modules in each stage of the subnetwork. Under this strategy, the transformation modules in each stage share the same parameters like applying a fixed solver iteratively for the blurred features. As shown in Fig. 3(c), the structure within one encoder stage of the subnetwork evolves from (b) to (c), in which the same module is used iteratively for nonlinear transformation. Formally, the function in each subnetwork is defined as

$$\mathbf{I}_i = \text{Net}_i(\mathbf{B}_i, \mathbf{I}_{i-1\uparrow}; \theta_i, \eta), \quad (1)$$

where Net_i is the subnetwork in the i -th scale with scale-independent parameters θ_i and scale-shared parameters η . In the i -th scale, the current blurred image \mathbf{B}_i and the upsampled restored sharp image at the $(i-1)$ -th scale $\mathbf{I}_{i-1\uparrow}$ are taken as input. The sharp image \mathbf{I}_i at this scale is pro-

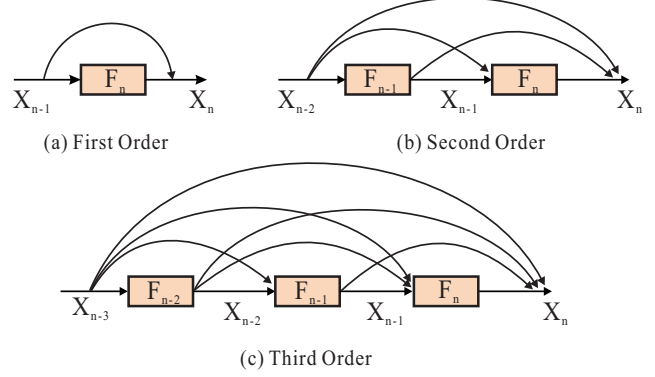


Figure 4. Higher-order residual functions result in nested skip connections.

duced. It is fed into the $(i+1)$ -th scale as the input for progressive restoration at next scale.

3.2. Nested Skip Connections

He *et al.* [7] validated that fitting the residual mapping rather than the desired mapping is much easier to optimize. Nah *et al.* [19] and Tao *et al.* [33] both chose ResBlocks as the internal building blocks for the blur-to-sharp feature transformation. Specifically, a ResBlock [19] is defined as

$$x_n = x_{n-1} + \mathcal{F}_n(x_{n-1}), \quad (2)$$

where x_{n-1} , x_n and \mathcal{F}_n are the input, output and the residual function of the n -th residual unit. We refer this as the first-order residual as shown in Fig. 4(a). If we assume the input x_{n-1} is also produced by another first-order residual function, we can put it into Eq. (2). Empirically, fitting the residual of residuals is easier than the original residual mapping. The second-order residual function is formulated as

$$x_n = x_{n-2} + \mathcal{F}_{n-1}(x_{n-2}) + \mathcal{F}_n(x_{n-2} + \mathcal{F}_{n-1}(x_{n-2})). \quad (3)$$

As shown in Fig. 4(b), there are 3 skip paths with one intersection in contrast to 2 short-term skip connections in stacked 2 ResBlocks. We further expand the second-order residual function to the third-order one as

$$x_n = x_{n-3} + \mathcal{F}_{n-2}(x_{n-3}) + \mathcal{F}_{n-1}(x_{n-3} + \mathcal{F}_{n-2}(x_{n-3})) + \mathcal{F}_n(x_{n-3} + \mathcal{F}_{n-2}(x_{n-3}) + \mathcal{F}_{n-1}(x_{n-3} + \mathcal{F}_{n-2}(x_{n-3}))). \quad (4)$$

Fig. 4(c) shows the third-order residual function. The recursion can be carried on to derive even higher-order residual functions. As shown in Fig. 4, these functions turn out to be a nested connected structure visually similar to DenseNet [8]. However, the difference is in two aspects. First, the skip connection here indicates feature summation instead of

channel concatenation. Second, the number of direct connections here is $\frac{(L+1)(L+2)}{2}$, with $(L+1)$ more links at the end of the last convolution layer compared with DenseNet.

Higher-order residual functions can be grouped into a nested module, to improve flow of information and better tackle gradient-vanishing issues throughout the network. Although the stacked ResBlocks in [19, 33] have many short-term skip connections, it simply stacks the first-order residual functions. Differently, our nested module models higher-order residual functions, which are capable of complex representation ability and easier to optimize. We use this nested module to replace the stacked ResBlocks for nonlinear transformation in different stages of our encoder-decoder subnetwork.

3.3. Network Architecture

Following [19] and [33], we utilize 3 scales in pursuing the “coarse-to-fine” strategy. Thus, three encoder-decoder subnetworks are stacked with independent feature extraction and reconstruction, and shared nonlinear transformation modules. Different from using kernel size 5×5 [19, 33], we use kernel size 3×3 to control the model size since 2 layers with 3×3 kernel can cover the same receptive fields as one layer with 5×5 kernel and it saves around 25% of the parameters.

By default, each nonlinear transformation module consists of 4 processing units, each composed of 2 convolution layers. The feature extraction and reconstruction modules are implemented as one convolution or transposed convolution layer respectively. This default setting aims at covering similar receptive fields to that of [33]. In each stage of the encoder-decoder subnetwork, our model has 17 convolution layers with kernel size 3×3 .

Given N training pairs of blurred and sharp images in S scales $\{B_i^k, L_i^k\}$, we minimize the Mean Squared Error (MSE) between the restored images and ground truth at each scale over the entire training set as

$$\mathcal{L}(\theta, \eta) = \frac{1}{2N} \sum_{k=1}^N \sum_{i=1}^S \frac{1}{T_i} \|F_i(B_i^k; \theta_i, \eta) - L_i^k\|_2^2, \quad (5)$$

where B_i^k and L_i^k are the blurred and ground truth images in the i -th scale respectively. θ_i denotes the scale-independent parameters, and η is the scale-shared parameter. The loss at each scale is normalized by the number of pixels T_i .

4. Experiments

Datasets Unlike generating blurred images by convolving blur kernels with sharp images, Nah *et al.* [19] synthesized realistic blurred images by averaging consecutive frames in a high-speed video. The released GoPro dataset contains 2,103 pairs for training and 1,111 pairs for evaluation. As shown in Fig. 5, there exist flaws in some of the ground

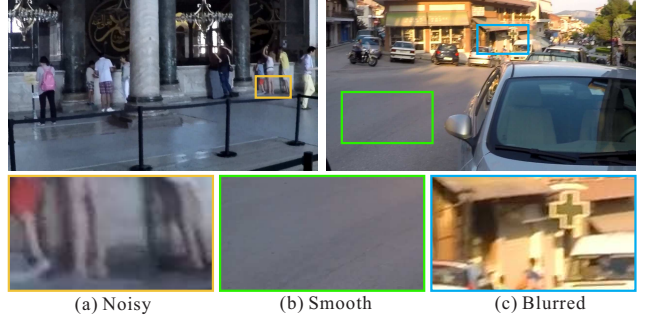


Figure 5. Several flaws exist in the ground truth sharp images in the GoPro training dataset.

truth sharp images in the GoPro training set, including severe noise, large smooth region, and significant image blur. To improve the training performance, we establish a new dataset following the procedures of [19] using GoPro Hero6 and iPhone7 at 240 fps.

We stick to 3 guidelines in collecting the videos to overcome the flaws. First, the camera is steady and we avoid recording high-speed vehicles or objects to ensure no camera motion or object motion exists in sharp frames. Second, we record outdoor videos in the daytime to guarantee a low noise level. Third, we only sample the scenes with enough details, and avoid large smooth regions such as sky or constant background. Under these guidance, we collect 5,290 blurred/sharp image pairs. This new dataset complements the GoPro dataset [19] to help dynamic scene deblurring research. Unless otherwise stated, the quantitative results in the following are based on the GoPro training dataset [19] for fair comparison.

Implementation We implement our algorithm by TensorFlow on a PC with Intel Xeon E5 CPU and an NVIDIA P40 GPU. During training, a 256×256 region from the blurred and ground truth images at the same location are randomly cropped as the training input. The batch size is set to 16 during training. All weights are initialized using Xavier method [4]; biases are initialized to 0. The network is optimized using Adam method [14] with default setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is initially set to 0.0001, exponentially decayed to 0 using power 0.3. According to our experiments, 4,000 epochs are sufficient for all the networks to converge.

4.1. Effectiveness of Parameter Selective Sharing

To demonstrate the effectiveness of the proposed parameter selective sharing scheme, we compare the proposed model (Model **SE Sharing**) with the parameter independence scheme and parameter sharing scheme. The parameter sharing scheme (Model **Sharing**) is implemented following [33]. The parameter independence scheme (Model **Indep.**) has the same network structure with Model **Shar-**

ing, but with independent parameters in each scale. For the selective sharing scheme, we use independent feature extraction and reconstruction modules, and shared nonlinear transformation module across different scales. Model **SE Sharing** employs intra-scale parameter sharing, with shared nonlinear transformation modules in each stage of the encoder-decoder subnetwork. We also test the strategy without intra-scale parameter sharing (Model **SE Sharing w/o IS**), where the 2 nonlinear transformation modules have different parameters.

Model	Indep.	Sharing	SE Sharing w/o IS	SE Sharing
Param	14.72M	4.91M	5.42M	2.84M
PSNR	30.65	30.79	30.97	30.92
SSIM	0.9369	0.9389	0.9426	0.9421

Table 1. Quantitative results for different parameter strategies.

The quantitative results are shown in Table 1, from which we obtain important observations. First, the parameter sharing scheme (Model **Sharing**) is indeed better than parameter independence scheme (Model **Indep.**) with higher performance and fewer parameters. Second, independent feature extraction and reconstruction modules can help further enhance the system compared with parameter sharing scheme. Third, the intra-scale parameter sharing (Model **SE Sharing**) yields comparable performance with the one without intra-scale parameter sharing (Model **SE Sharing w/o IS**). Note it is only with around half of the parameters.

4.2. Effectiveness of Nested Skip Connections

To demonstrate the effectiveness of the nested skip connections, we compare this structure with several baseline structures. For fair comparison, all the models have 8 convolutions in each stage of the encoder-decoder subnetwork. Model **Plain** simply stacks 8 convolution layers. Model **ResBlock** uses 4 ResBlocks in each module. Model **DenseBlock** stacks 2 DenseBlocks following DenseNet [8]. Model **Nested** represents the proposed nested skip connection structure.

Model	Plain	ResBlock	DenseBlock	Nested
PSNR	29.84	30.76	28.85	30.92
SSIM	0.9248	0.9383	0.9109	0.9421

Table 2. Quantitative results for different module structures.

As shown in Table 2, model **ResBlock** performs better than model **Plain**. They both work better than model **DenseBlock** since the growth rate is set to a small value to make the output channels of the DenseBlock same as other structures. The table indicates that the proposed nested skip connection structure achieves better performance than others.

Model	Gong	Nah	Tao	Zhang	Ours	Ours+
PSNR	26.06	29.08	30.26	29.19	30.92	31.58
SSIM	0.8632	0.9135	0.9342	0.9306	0.9421	0.9478
Time	20min	3.1s	1.3s	1.4s	1.6s	1.6s

Table 3. Quantitative results on GoPro evaluation dataset.

4.3. Comparison with Other Deblurring Methods

We compare our method with recent state-of-the-art dynamic scene deblurring and non-uniform deblurring methods on the GoPro evaluation dataset quantitatively, as well as on more blurred images qualitatively. Sun *et al.* [31] and Gong *et al.* [5] both estimated the blur fields and use non-blind deconvolution method to recover the sharp image. Since the method of [5] can handle general motion rather than local linear motion [31], we only compare ours with the solution of [5]. Nah *et al.* [19] and Tao *et al.* [33] employed parameter independence and parameter sharing schemes respectively in building their deep networks. Zhang *et al.* [37] proposed an spatially variant RNN for dynamic scene deblurring. The quantitative results on GoPro evaluation dataset are listed in Table 3. As shown in the last column of Table 3, we also list the results trained on mixed GoPro and our dataset. The statistics demonstrate the advantages of adding our dataset for training.

Visual comparison on GoPro evaluation dataset is shown in Fig. 6. These results are generated by the model trained only on the default GoPro training dataset. To test the generalization ability of our model, we apply our best-performing model to more images. We collect synthetic blurred images from [15], download blurry images from Internet, and sample real blurred images. As shown in Fig. 7, our model generally produces better results than those of [19] and [33]. Our model handles non-uniform and highly dynamic scenes quite well compared with others, as shown in the close-ups from the first and second images in Fig. 7. On the third and fourth images in Fig. 7, our method successfully restores more recognizable text details than others.

5. Conclusion

In this work, we have analyzed the general principle of using parameters wisely in deblurring CNNs and proposed a parameter selective sharing scheme in contrast to parameter independence and sharing schemes. We have also proposed a new nested skip connection structure for the nonlinear transformation modules in the network. Besides, we have built a large blurred/sharp paired dataset towards training better models. By adopting the parameter selective sharing scheme, nested skip connection structure and our new training dataset, we have presented a new state-of-the-art in dynamic scene deblurring.

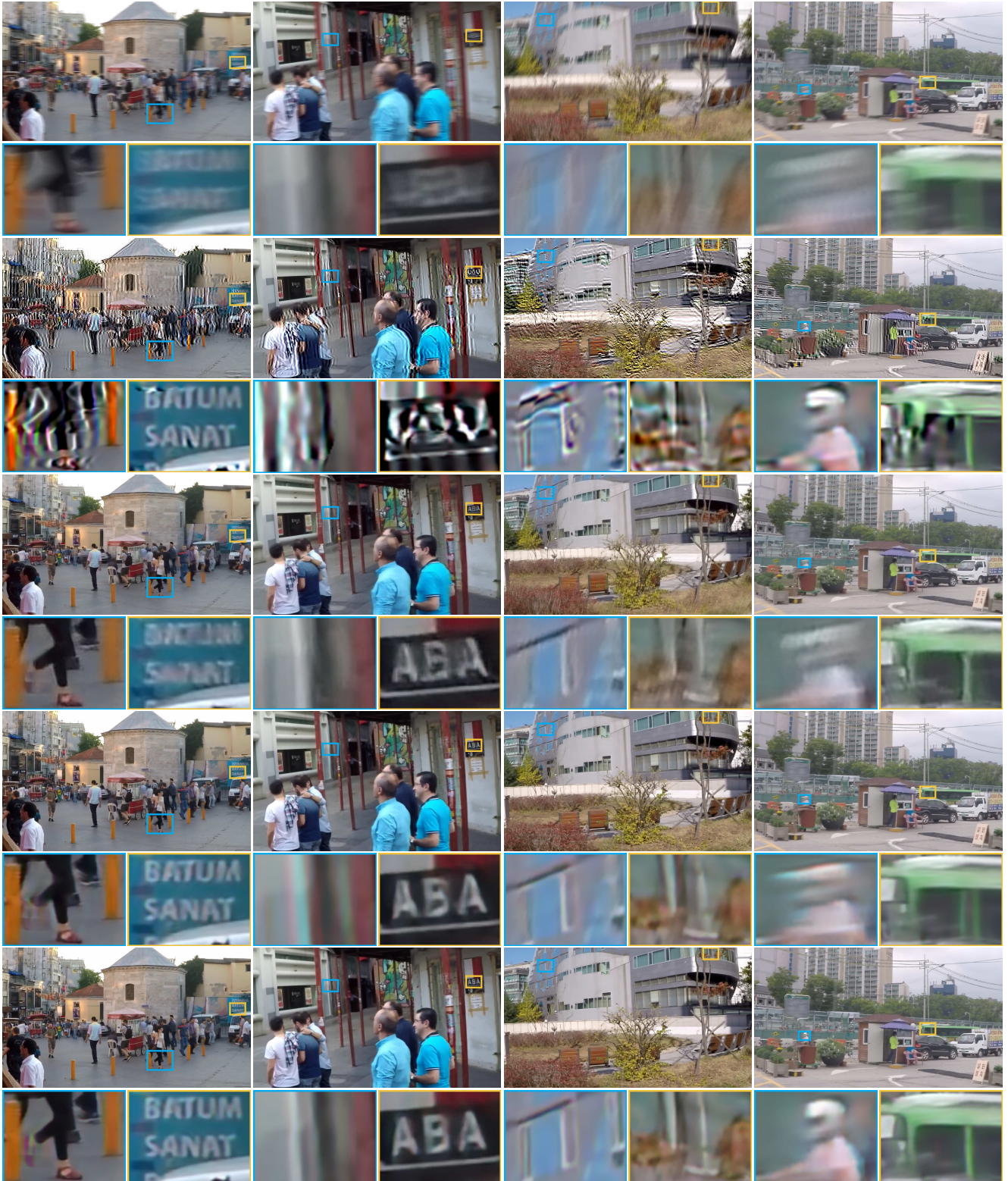


Figure 6. Visual comparison on GoPro evaluation dataset. From the top to bottom, we show input, results of Gong *et al.* [5], Nah *et al.* [19], Tao *et al.* [33] and ours (best view on screen).

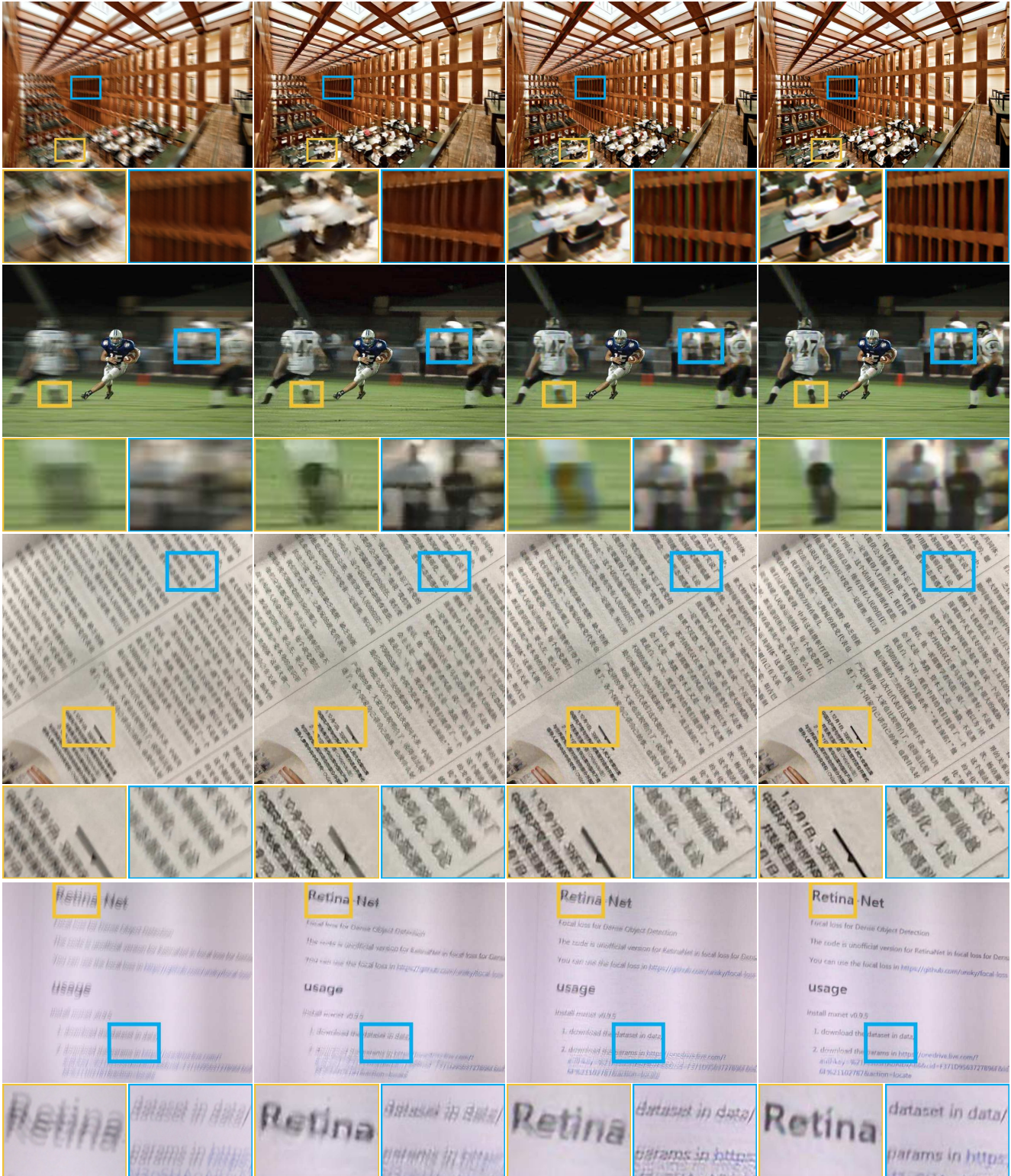


Figure 7. Visual comparison on more blurred images. The first image is from the synthetic dataset [15]. The second image is from the Internet. The third and fourth images are captured by our iPhone 7. The first column is the input image. The second column is generated by [19]. The third column is produced by [33]. The fourth column is our results trained on mixed datasets. Best viewed on screen.

References

- [1] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, 1998. [1](#)
- [2] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics*, 28(5), 2009. [1](#)
- [3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015. [3](#)
- [4] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTAT*, pages 249–256, 2010. [5](#)
- [5] D. Gong, J. Yang, L. Liu, Y. Zhang, I. D. Reid, C. Shen, A. Van Den Hengel, and Q. Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *CVPR*, pages 2319–2328, 2017. [1](#), [2](#), [6](#), [7](#)
- [6] M. Haris, G. Shakhnarovich, and N. Ukita. Deep backprojection networks for super-resolution. In *CVPR*, pages 1664–1673, 2018. [3](#)
- [7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. [1](#), [3](#), [4](#)
- [8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. [3](#), [4](#), [6](#)
- [9] Z. Hui, X. Wang, and X. Gao. Fast and accurate single image super-resolution via information distillation network. In *CVPR*, pages 723–731, 2018. [3](#)
- [10] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. [3](#)
- [11] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, pages 1637–1645, 2016. [3](#)
- [12] T. H. Kim, B. Ahn, and K. M. Lee. Dynamic scene deblurring. In *ICCV*, pages 3160–3167, 2013. [1](#), [2](#)
- [13] T. H. Kim and K. M. Lee. Segmentation-free dynamic scene deblurring. In *CVPR*, pages 2766–2773, 2014. [1](#), [2](#)
- [14] D. P. Kingma and J. Ba. A method for stochastic optimization. In *ICLR*, 2015. [5](#)
- [15] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang. A comparative study for single image blind deblurring. In *CVPR*, pages 1701–1709, 2016. [6](#), [8](#)
- [16] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. In *CVPR*, pages 3367–3375, 2015. [3](#)
- [17] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. [1](#)
- [18] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NIPS*, pages 2802–2810, 2016. [3](#)
- [19] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [20] J. Pan, Z. Hu, Z. Su, H.-Y. Lee, and M.-H. Yang. Soft-segmentation guided object motion deblurring. In *CVPR*, pages 459–468, 2016. [1](#), [2](#)
- [21] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *CVPR*, pages 2901–2908, 2014. [1](#)
- [22] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *CVPR*, pages 1628–1636, 2016. [1](#)
- [23] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014. [3](#)
- [24] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, pages 2482–2491, 2018. [3](#)
- [25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. [3](#)
- [26] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1439–1451, 2016. [1](#)
- [27] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3), 2008. [1](#)
- [28] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *NIPS*, pages 656–664, 2012. [3](#)
- [29] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In *NIPS*, pages 2377–2385, 2015. [3](#)
- [30] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring. In *CVPR*, pages 1279–1288, 2017. [3](#)
- [31] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, pages 769–777, 2015. [1](#), [2](#), [6](#)
- [32] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 3147–3155, 2017. [3](#)
- [33] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, pages 8174–8182, 2018. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [34] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *ICCV*, pages 4809–4817, 2017. [3](#)
- [35] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, pages 157–170, 2010. [1](#)
- [36] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, pages 1107–1114, 2013. [1](#)
- [37] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. Lau, and M.-H. Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *CVPR*, pages 2521–2529, 2018. [1](#), [2](#), [6](#)
- [38] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *CVPR*, pages 2472–2481, 2018. [3](#)
- [39] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011. [2](#)