LONG-TERM RESIDUAL BLENDING NETWORK FOR BLUR INVARIANT SINGLE IMAGE BLIND DEBLURRING

Sungkwon An\textsuperscript{1}, Hyungmin Roh\textsuperscript{1}, Myungjoo Kang\textsuperscript{2}

\textsuperscript{1}Computational Science and Technology, Seoul National University
\textsuperscript{2}Department of Mathematics, Seoul National University
sk_an@snu.ac.kr, hmroh.snu@gmail.com, mkang@snu.ac.kr

ABSTRACT

We present a novel, blind, single image deblurring method that utilizes information regarding blur kernels. Our model solves the deblurring problem by dividing it into two successive tasks: (1) blur kernel estimation and (2) sharp image restoration. We first introduce a kernel estimation network that produces adaptive blur kernels based on the analysis of the blurred image. The network learns the blur pattern of the input image and trains to generate the estimation of image-specific blur kernels. Subsequently, we propose a long-term residual blending network that restores sharp images using the estimated blur kernel. To use the kernel efficiently, we propose a blending block that encodes features from both blurred images and blur kernels into a low dimensional space and then decodes them simultaneously to obtain an appropriately synthesized feature representation. We evaluate our model on REDS, GOPRO and Flickr2K datasets using various Gaussian blur kernels. Experiments show that our model can achieve excellent results on each dataset.

1. INTRODUCTION

Herein, a novel, deep-learning-based, single image deblurring method that restores sharp images from arbitrary blurred images based on an estimated blur kernel is introduced. Many studies have been performed to solve the single image deblurring problem, and most previous approaches can be categorized into two groups: non-blind and blind methods. The non-blind method utilizes information from a blur kernel, whereas the blind method restores a sharp image without using any information from a blur kernel.

Although recent studies based on both methods have demonstrated good results, the two methods still present some limitations. First, when applying the non-blind method, known blur kernels are assumed. However, blur kernels are not known in most low-quality images; therefore, it is difficult to apply the non-blind method to real-world problems. Meanwhile, because the blind method does not require information regarding the blur kernel, it can be expanded limitlessly to a real-world problem.

Most state-of-the-art single image deblurring methods have adopted the blind method, which uses only a blur image to apply to the real-world. The blind method was developed owing to the difficulty in obtaining information regarding a blur kernel from arbitrary images. Nah et al. \cite{10} proposed a deblurring network called DeepDeblur, which does not require knowledge regarding blur kernels. They opted not to include that information because incorrectly predicted kernels may introduce unintended artifacts to the results. However, if kernel information is accurately predicted, the performance can be further enhanced. Gu et al. \cite{4} proposed a super-resolution method that predicts the feature of low-dimensional projected blur kernels using principal component analysis (PCA). However, they could not fully utilize information regarding the blur kernel because they reduced the kernel dimension. Cornill`ere et al. \cite{2} estimated an adaptive downscaling kernel from a low-resolution image and used that information to generate a corresponding high-resolution image.

We herein propose a long-term residual blending (LTRB) network to address the limits of previous methods and further improve the results. Our model comprises two stages. The first is a kernel estimation network that predicts the blur kernel of a blurred image. The second is an LTRB network that restores sharp images using information regarding the estimated kernel. To reconstruct images accurately, we first estimate suitable blur kernels by analyzing the blurred images. Subsequently, our LTRB network restores sharp images by analyzing the detail and structure of the images using the information from predicted blur kernels. The architectural design of the LTRB network is based on non-blind methods, in that estimated blur kernels are used. However, when combined with our kernel estimation network, the LTRB network can be applied to arbitrary images as a blind method. Furthermore, we propose the blending block and long-term skip connection that enable blurred images to be analyzed without any loss in coarse features from original images when the LTRB network deepens. Consequently, we were able to restore in-
formation from both coarse and fine-grained features.

We created datasets using various anisotropic Gaussian kernels to validate our model on datasets that are close to real-world images rather than using existing diversity-limited datasets. Our model demonstrated excellent performance on data constructed for our purposes. We present our method to predict the blur kernel in Section Proposed Methods, where we also introduce a state-of-the-art LTRB network for single image deblurring problems.

Our contribution is as follows:

- We introduce a novel LTRB network that can analyze coarse features from blurred images in a deep network and utilize estimated blur kernels.
- We propose a blending block to improve the understanding of the image deblurring process and the method to use blur kernel estimation efficiently.
- We propose a long-term skip connection that allows our network to consider all progressively analyzed features from fine to coarse features.
- By combining the kernel estimation and LTRB network, we introduce an end-to-end deblurring network pipeline that is not only applicable to real-world problems (like blind methods), but also yield accurate results (like non-blind methods).

2. RELATED STUDIES

In this section, we provide a brief review of some non-blind and blind approaches for solving the image deblurring problem.

2.1. Non-blind methods

The non-blind method of restoring a sharp image using blur kernel information has been investigated continuously. Researchers have attempted to combine recent deep learning techniques and the traditional Wiener filter [18] to deconvolve images using blur kernels. Kruse et al. [6] proposed an FFT-based non-blind deblurring method that utilizes an improved Wiener filter for a specified blur kernel and a CNN-based term as input. However, the circular blur assumption is required for an efficient FFT-based optimization. Wang et al. [17] proposed a method to manage different types of blur kernels in a single model. They estimated the residuals from a CNN model after predeconvolving blur images using the regularized Wiener method.

Furthermore, Zhang et al. [19] used a deep learning technique to perform iterative de-noising and deblurring using the FCNN with a deconvolution module. Ren et al. [14] approximated an arbitrary pseudo-inverse kernel as multiple matrices and initialized the convolutional parameters of a deblurring network using these matrices. Their initialization method enabled the network to avoid poor local minima and yield a favorable deblurring performance.
2.2. Blind methods

More recent works focus on the blind method, which is applicable to any blurred image even if no information exists regarding its blur kernel. Because the blind method does not require information regarding the blur kernel in the image restoration process, it can be extended to real-world problems more effectively than the non-blind method.

Because the application of multiscale architectures to various image processing tasks have demonstrated good performances, they have also been applied to the deblurring task. Nah et al. [10] proposed a multiscale CNN called DeepDeblur, which restores a sharp image step by step by applying networks with the same structure from the coarsest level to a finer level and concatenating sharp features from the previous network. Similarly, Tao et al. [16] restored sharp images at multiple resolutions through a coarse-to-fine scale analysis. In a different attempt, Kupyn et al. [7] proposed DeblurGAN with a conditional GAN framework and content loss. Additionally, they proposed DeblurGAN-v2 [8] with an FPN applied to a generator and global and local discriminators. Shen et al. [15] proposed a triple-branch encoder–decoder structure that sharpens the foreground and background and analyzed them globally.

For the blind image enhancement method, researchers have attempted to estimate the degradation kernel and restore the image, similar to the non-blind method. Pan et al. [13] applied motivation to deblurring by obtaining sparse dark pixels in the blurring process. Blur kernels were estimated by calculating the dark channel using a linear method; subsequently, the estimated kernel and the existing deblurring method were used to obtain a restored image. In a slightly different manner, Pan et al. [12] proposed a method to estimate blur kernels by exploiting reliable edges from blurred images and removing outliers from intermediate latent images. Furthermore, a robust energy function was proposed to develop an advanced deblurring algorithm. Cornillère et al. [2] trained a kernel discriminator to predict whether an inappropriate blur kernel had been applied after generating a super-resolution image. Moreover, by minimizing the error predicted by the kernel discriminator, they generated a suitable degradation kernel and corresponding HR images.

It is clear that various methods have been proposed to estimate and use degradation kernels in the image enhancement method. Meanwhile, some researchers have focused on blur kernel estimation. Bell–Kligler et al. [1] proposed KernelGAN to obtain a suitable downscaling kernel to obtain an effective image-specific super-resolution. Their method analyzes the patch distribution of a low-resolution image through adversarial learning to generate a downscaling kernel.

3. PROPOSED METHOD

In this section, we provide the problem formulation and the architectural design of our proposed networks for blur kernel estimation and the image deblurring problem.

3.1. Problem Formulation

In the basic image deblurring problem, a sharp image $S$ and a blurred image $B$ are expressed as follows:

$$B = k \ast S + n,$$

where $k$ is the blur kernel, $n$ the additive noise, and $\ast$ the convolution operator. Kernel $k$ provides various blur effects, such as motion blur and Gaussian blur depending on its type.
In this problem, most methods estimate a deblurred image using a network that renders a blurred image sharp. We extend the deblurring problem to the problem of predicting kernel $k$, which is the cause of blurring, and use it to obtain a sharp image $S$. Obtaining the kernel that causes a blur image in the deblurring problem and using that information will not only yield more advanced results compared with existing methods, but will also enable us to understand the deblurring process. Equations (2) and (5) represent the kernel estimation and deblurring mechanism, respectively.

$$\theta_g^* = \arg \min_{\theta_g} || G (B, \theta_g) * S - B ||$$  \hspace{1cm} (2) \\
$$B \approx G (B, \theta_g^*) * S = k^* * S,$$  \hspace{1cm} (3)

where $G$ is the kernel estimation network, $\theta_g$ denotes the parameters of $G$, and $k^*$ is the estimated blur kernel.

$$\theta_n^* = \arg \min_{\theta_n} || \mathcal{N} (B, k^*, \theta_n) - S ||$$  \hspace{1cm} (4) \\
$$= \arg \min_{\theta_n} || \mathcal{N} (B, G (B, \theta_g^*), \theta_n) - S ||,$$  \hspace{1cm} (5)

where $\mathcal{N}$ is the deblurring network, and $\theta_n$ denotes the parameters of $\mathcal{N}$.

We focused on deblurring images blurred by various Gaussian kernels. Additional experimental methods and results of motion blur kernels obtained using our method are provided in the Experiments section.

### 3.2. Blur Kernel Estimation Network

By estimating the blur kernel during the deblurring process, we can understand the blur information and obtain more accurate deblurred results. KernelGAN estimates the degradation kernel that generates a low-resolution image by utilizing GAN’s ability to learn data representation. We adopted KernelGAN’s pipeline to estimate the blur kernel of a blurred image. The architecture is shown in Figure 3. KernelGAN estimates the downscaling kernel by learning the internal distribution of patches from low-resolution images. Its network contains a generator that downscales the input image and a discriminator that determines the distribution of patches in the input image.

We modified the KernelGAN structure by replacing the downsampling generator with our blurring generator such that the model fitted our task. We trained the discriminator to distinguish between the blur information of the input image and the fake blur image generated by the generator, whereas the blurring generator was trained to generate a fake blur image that can share the blur pattern of the input image such that it can deceive the discriminator. Consequently, we were able to extract the blur kernel of the input image using our blurring generator.

### 3.3. LTRB Network

Figure 2 illustrates the pipeline of our LTRB network. We opted not to reduce the dimension of the feature maps from the input images by avoiding the usage of pooling layers to maintain the maximum amount of spatial information from blurred images. Hence, when applying the kernel feature to the blur image feature as a condition, an imbalance problem occurs for each feature size. Therefore, we introduced a blending block, which comprises an encoder and a decoder, inspired by the conditional autoencoder.

We projected feature maps from the blurred image and the estimated kernel into lower-dimensional spaces, using our blending encoder and kernel encoder, respectively. Subsequently, we concatenated these two features and placed them into our blending decoder to obtain an appropriately synthesized feature representation.

The output feature of the blending block was decomposed into a large number of channels. As shown in many studies, channel attention demonstrated good performances in convolutional networks with a large number of feature channels. In the LTRB network, we applied channel attention after the blending block to rescale important features.

Because the LTRB network is composed of a large number of blending blocks, information extracted from the coarse level may be forgotten when they are passed through deep layers. We avoided information loss by connecting the coarse level features to the middle of the blending blocks using a long-term skip connection. Using this connection, we solved the vanishing information problem and obtained good results for our deeply stacked network.

Many deblurring methods exist that utilize residual blocks with skip connections. Despite the usage of these residual blocks, the consideration of fine features in the image is limited. The feature of the blurred image in the early stage of the network shows many fine features, and the long-term skip connection is proposed, such that these features can be considered deep in the network. This connection enables our network to consider all the progressively analyzed features from the fine to coarse features by repeatedly placing the initial blur image features together in all blending blocks.

---

![Fig. 3: Kernel Estimation Network Architecture](image-url)
Fig. 4: Sixteen blur kernels. We used 4 Gaussian kernels with various variances and 12 anisotropic Gaussian kernels with various rotation angles.

4. EXPERIMENTS

In this section, we present the experimental condition and results by comparing the performances of our model and other models on three other datasets.

4.1. Implementation Details

We trained our model on REDS [11], GOPRO [10], and Flickr2K [9] datasets. From the REDS dataset, we used 24,000 images for training and 3,000 for testing. For the GOPRO dataset, we used 2,103 images for training and 1,111 for testing. To train our model on the Flickr2K dataset, we randomly cropped HR images to increase the number of data and used 20,000 and 3,000 images for training and testing, respectively.

Our model was trained on Titan X and Titan RTX. We used the ADAM optimizer with a learning rate of $2 \times 10^{-4}$ and trained for 200k iterations.

4.2. Data Generation

Most existing deblurring datasets tend to focus on the motion blur. Because we aim to train and test our model on a Gaussian blur, we created new datasets using our own blur kernels. Among the existing motion blur datasets, we used sharp images from REDS and GOPRO, and we used high-resolution images from the Flickr2K Super-Resolution dataset. Whereas REDS and GOPRO are composed of daily images from various scenes, Flickr2K contains detailed texture and delicate patterns. Therefore, these datasets are suitable for evaluating the performances of deblurring networks.

We used $17 \times 17$ isotropic Gaussian kernels and anisotropic Gaussian kernels, both of which provide Gaussian and motion blur effects to sharp images. Each Gaussian blur kernel was created with different variance sizes and anisotropic Gaussian blurs of different rotations.

Figure 4 illustrates the blur kernels we used in this study. Each kernel was randomly selected from 16 blur kernels and applied to each sharp image. Figure 5 shows the blurred images generated by our blur kernels and the corresponding estimated kernels predicted by the kernel estimation network. Therefore, we were able to obtain data pairs comprising a blur image, a sharp image, a blur kernel, and an estimated kernel, which we used to train our model.

4.3. Kernel Estimation

To estimate the blur kernel in blind deblurring, we used our kernel estimation network shown in Figure 3. The blurred image we created was placed in the kernel estimation network to obtain the corresponding $17 \times 17$ blur kernel prediction result. Figure 5 shows the kernel estimation results obtained using our method and several other methods. As shown, our method approximates the kernel features, such as the rotation angle, similar to other methods [12] [13]. During training, we first predicted the blur kernel of the blurred image and then delivered the predicted kernel to the blending block in the deblurring model.

For a fair comparison, we tested our model with the estimated kernel corresponding to the test blurred images, whereas ground-truth kernels were provided to other non-blind models because they required the blur kernel information.

4.4. Deblurring Results

We compared some methods that demonstrated excellent results with non-deep learning and deep learning methods. Fast-deconv [5] and EPLL [20] of the non-deep learning method
Table 1: Comparison of deblurring result on REDS, GOPRO, and Flickr2K datasets. Average PSNR and SSIM are provided with the best values marked in bold.

| Method                  | REDS | GOPRO | Flickr2K |
|-------------------------|------|-------|----------|
|                         | PSNR | SSIM  | PSNR     | SSIM  | PSNR     | SSIM  |
| Blurred Image           | 25.9908 | 0.6420 | 28.9878 | 0.7629 | 27.6748 | 0.6750 |
| fastdeconv              | 28.4818 | 0.7644 | 33.0048 | 0.8734 | 30.2462 | 0.7838 |
| EPLL                    | 30.3059 | 0.8109 | 35.0216 | 0.9058 | 31.9507 | 0.8261 |
| Outlier-Deblur          | 28.1655 | 0.7652 | 30.0099 | 0.8432 | 27.8995 | 0.7468 |
| Dark-Channel            | 27.6806 | 0.7076 | 31.0967 | 0.8433 | 28.2186 | 0.7199 |
| DeepDeblur              | 32.0374 | 0.8690 | 36.0505 | 0.9291 | 32.0912 | 0.8514 |
| SRN                     | 32.2751 | 0.8777 | 36.4768 | 0.9365 | 33.0973 | 0.8623 |
| DeblurGAN-v2            | 28.9220 | 0.8059 | 32.5397 | 0.8709 | 28.5913 | 0.7860 |
| Ours w/o Blending Block | 32.7657 | 0.8978 | 36.6472 | 0.9351 | 32.5558 | 0.8552 |
| Ours w/o LTS Connection | 32.7150 | 0.8786 | 36.7864 | 0.9384 | 33.2392 | 0.8630 |
| Ours                    | **33.4282** | **0.8938** | **37.2848** | **0.9412** | **33.5814** | **0.8729** |

and DeepDeblur [10], SRN [16], and DeblurGAN-v2 [8] of the deep learning method were compared. Additionally, we compared Dark-Channel [13] and Outlier-Deblur [12], which performed deblurring using the estimated kernel, similar to ours. Each method was trained and tested using the published official code. In addition, we conducted ablation studies to evaluate the effect of our long-term skip connection and blending blocks using the estimated kernel. Additionally, we applied our model to the motion deblurring task to examine the effect. We first removed the blending block from our model to uncover the effect of the estimated blur kernel in deblurring, such that the kernel information was not used in our model. Subsequently, we tested our model without a long-term skip connection, the feature of which we expect to gradually utilize from coarse to fine. We used the PSNR and SSIM to evaluate our results.

Table 1 shows our experimental results. The deep learning methods indicated better results than the non-deep learning methods, and our proposed model demonstrated the best performance for all indicators. The SRN demonstrated the second best results followed by DeepDeblur with a slight difference. Because sharp images composing the GOPRO dataset were captured using a portable action camera called GOPRO, their resolutions were low. Therefore, the experiments on GOPRO yielded higher PSNR and SSIM values compared with experiments on other datasets.

As shown in Table 1, the model without the blending block and long-term skip connection demonstrated a slightly lower PSNR and SSIM than our complete model. This shows that the model was better or of a similar level with a margin smaller than the SRN, which ranked top-2 in our comparative experiment. Based on this experiment, it was clear that the components of our proposed model enabled this good performance.

Figure 6 shows detailed comparisons of qualitative results of different models on REDS, GOPRO, and Flickr2K datasets. Whereas DeepDeblur and SRN indicated better performances compared with other models, some details were still distorted and restored differently from the ground-truth. Our model, however, successfully restored such details in most cases.

4.5. Additional Experiments

Table 2: Comparison of motion deblurring results on REDS dataset

| Method                  | REDS   |
|-------------------------|--------|
|                         | PSNR   | SSIM  |
| Blurred Image           | 23.5828 | 0.5117 |
| DeepDeblur              | 30.5306 | 0.8191 |
| DeblurGAN-v2            | 26.1420 | 0.6492 |
| Ours                    | **30.7839** | **0.8237** |

We can use our deblurring network to solve motion and Gaussian deblurring problems in a similar process: estimate the motion kernel, or the motion flow, from the blurred image and then restore the sharp image. In particular, to solve the motion deblurring problem, we estimated the pixel-wise motion flow, such that our model can remove pixel-wise heterogeneous motion blurs.
Fig. 6: Visual comparison of deblurring results. In the blurred image, detailed patterns were severely damaged and difficult to recognize (a) checkered pattern of the net; (b) horizontal line of the roof; (c) outline of letters and texture of the metal plate). Most methods succeeded in partially restoring such patterns but more than half remained blurry. Using the blur kernel information, our method successfully restored the largest portion of damaged details and appeared the most similar to the ground-truth compared to other methods.
Gong et al. introduced a method for generating a dataset by translating each pixel in the $x$-, $y$-, and $z$-axes, assuming that a motion blur can be expressed as a pixel-size motion flow [3]. They estimated pixel-wise motion flows using a modified FCN network. For convenience, we first assumed that the pixel-wise motion flow was that assumed by Gong et al. Subsequently, we trained and evaluated our LTRB network using that pixel-wise motion flow in a non-blind manner. For the dataset, we generated blurred images from sharp images in the REDS dataset using the blurring method proposed by Gong et al.

To create motion flows in our model and for the motion deblurring task, we modified the structure of the blending encoder and decoder modules in the blending blocks from the LTRB network. Because motion flows are of the same size as the blurred image, unlike the Gaussian blur kernel, we concatenated the motion flow and the blurred image and then placed them into the blending block as a condition.

Figure 7 and Table 2 show a comparison of the results obtained using our model and other state-of-the-art methods. In terms of the PSNR and SSIM, Table 2 shows that our model performed better than the other methods. Based on the results of the reconstructed image in Figure 7, our model and DeepDeblur performed well together; however, the result of DeepDeblur showed some artifacts and insufficient details. Our model not only performed well on the deblurring task, but also demonstrated good results in the motion deblurring domain. We expect our model to perform better if we further change the design of the encoder that projects the features of motion flow to render our model more optimized for the motion deblurring task. However, this will only be attempted in future research.

5. CONCLUSION

Herein, we presented a novel deblurring method. Using our proposed blending block and long-term skip connection, we improved the understanding of the image deblurring process and the method to use blur kernel estimation efficiently. Our model not only demonstrated excellent results in the deblurring task, but it also indicated the potential to be applied to other image enhancement tasks such as motion deblurring.

References

[1] S. Bell-Kligler, A. Shocher, and M. Irani. Blind super-resolution kernel estimation using an internal-gan. In Advances in Neural Information Processing Systems, pages 284–293, 2019.

[2] V. Cornillère, A. Djelouah, W. Yifán, O. Sorkine-Hornung, and C. Schroers. Blind image super-resolution with spatially variant degradations. ACM Transactions on Graphics (TOG), 38(6):1–13, 2019.

[3] D. Gong, J. Yang, L. Liu, Y. Zhang, I. 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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2319–2328, 2017.

[4] J. Gu, H. Lu, W. Zuo, and C. Dong. Blind super-resolution with iterative kernel correction. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1604–1613, 2019.

[5] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In Advances in neural information processing systems, pages 1033–1041, 2009.

[6] J. Kruse, C. Rother, and U. Schmidt. Learning to push the limits of efficient fft-based image deconvolution. In Proceedings of the IEEE International Conference on Computer Vision, pages 4586–4594, 2017.

[7] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8183–8192, 2018.

[8] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In Proceedings of the IEEE International Conference on Computer Vision, pages 8878–8887, 2019.

[9] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 136–144, 2017.
[10] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[11] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. M. Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019.

[12] J. Pan, Z. Lin, Z. Su, and M.-H. Yang. Robust kernel estimation with outliers handling for image deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2800–2808, 2016.

[13] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1628–1636, 2016.

[14] W. Ren, J. Zhang, L. Ma, J. Pan, X. Cao, W. Zuo, W. Liu, and M.-H. Yang. Deep non-blind deconvolution via generalized low-rank approximation. In Advances in Neural Information Processing Systems, pages 297–307, 2018.

[15] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao. Human-aware motion deblurring. In Proceedings of the IEEE International Conference on Computer Vision, pages 5572–5581, 2019.

[16] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8174–8182, 2018.

[17] R. Wang and D. Tao. Training very deep cnns for general non-blind deconvolution. IEEE Transactions on Image Processing, 27(6):2897–2910, 2018.

[18] N. Wiener. Extrapolation, interpolation, and smoothing of stationary time series. The MIT press, 1964.

[19] J. Zhang, J. Pan, W.-S. Lai, R. W. Lau, and M.-H. Yang. Learning fully convolutional networks for iterative non-blind deconvolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3817–3825, 2017.

[20] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In 2011 International Conference on Computer Vision, pages 479–486. IEEE, 2011.