2023 IEEE 9th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) | 979-8-3503-4452-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAMSAP58249.2023.10403451

UNCERTAINTY AWARE KERNEL ESTIMATION AND EDGE PRESERVING ATTENTION FOR BLIND IMAGE DEBLURRING

Nithin Gopalakrishnan Nair and Vishal M. Patel

Dept. of Electrical and Computer Engineering, Johns Hopkins University, MD, USA
{ngopala2, vpatel36}@jhu.edu

ABSTRACT

Blind deconvolution is a challenging problem because of its ill-posed nature. Most existing blind deconvolution techniques are based on classical methods and utilize a maximum a posterior (MAP) framework to estimate clean images and blur kernels. Very recently, a method that utilizes the Deep Image Prior (DIP) principle has been proposed. This method uses two generative networks to model the deep priors of clean image and blur kernel. But this method fails for complex kernels, and estimates erroneous kernels, hence leading to ringing artifacts in the reconstructed image. To address this issue and estimate better kernels, we introduce a Bayesian uncertainty guided kernel estimation technique. Also, to improve the quality of the reconstructed images, we present a new type of edge-preserving attention. We perform evaluations on several benchmark datasets to show the performance improvement obtained by our network.

1. INTRODUCTION

Images captured by hand-held cameras often suffer from blurring due to camera motion or movements in the captured scene. Assuming that the scene captured is static, photographs captured by hand-held cameras with large exposure times are prone to blur degradation due to camera shake. The removal of blur from degraded images is of significant interest to photographers. Moreover, multiple computer vision applications require clean images for their proper functioning[1, 2]. Hence there exists a need to restore images degraded by camera motion. In normal scenarios, the distortion due camera shake can be modelled as a convolution operation of the underlying clean image with a space invariant blur kernel[3, 4, 5, 6] and can be expressed as ,

$$y = k * x + n, \tag{1}$$

where, y is the blurry image, x is the latent clean image, k is the blur kernel and n is an additive white Gaussian noise. The operator $*$ denotes convolution. The task of retrieving x given y, without the knowledge of k is referred to as the blind deblurring (or deconvolution) problem. Blind image deconvolution is challenging because of its highly ill-posed nature. Moreover, we have to retrieve the clean latent image x as well as the blur kernel k from the degraded observation y. Very recently, Ren *et al.*.[3] proposed a method that uses the Deep Image Prior [7] concept and utilize two generative models. This method performs an unconstrained neural optimization to estimate the clean latent image and blur kernel. But the restored kernels are prone to small errors after optimization and initialization plays a major role on the quality of kernel predicted. It is well known that small uncertainties in the kernel estimation cause artifacts and ringing effects in the restored images [8, 9].

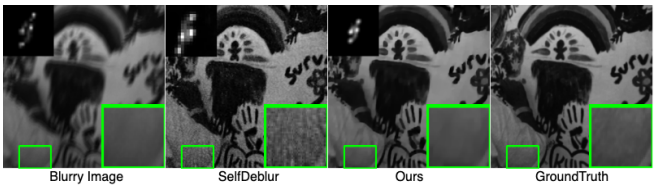

Fig. 1: A visualization of estimated uncertain kernels and resulting artifacts caused by them.

Hence, to improve the quality of generated kernels, we propose a new method by using the concept of uncertainty to the process of kernel estimation. For this, we create an uncertainty-aware kernel estimation network by estimating the epistemic uncertainty [10, 11] of the predictions of a base network. We consider the kernel estimation network as a Bayesian neural network by including dropout layers as in Gal et al.[11]. We then predict multiple kernels by utilizing Monte-Carlo sampling with dropout and perform approximate inference of the true distribution of the predicted kernels. To get a refined estimate of the kernel, we use a refinement network that takes the first and second-order moments of the approximated distribution as input. We also propose a novel loss, devised using the second-order moments of the kernel distribution and constrain the space of the generated kernel.

Artifacts often arise due to imperfect deconvolution operation or due to the presence of kernel errors [8, 9]. Hence a second stage that could correct the estimated latent image would be an important task in blind deconvolution. Consider the features of the penultimate layer of the image pre-

This research was supported by NSF CAREER award 2045489.

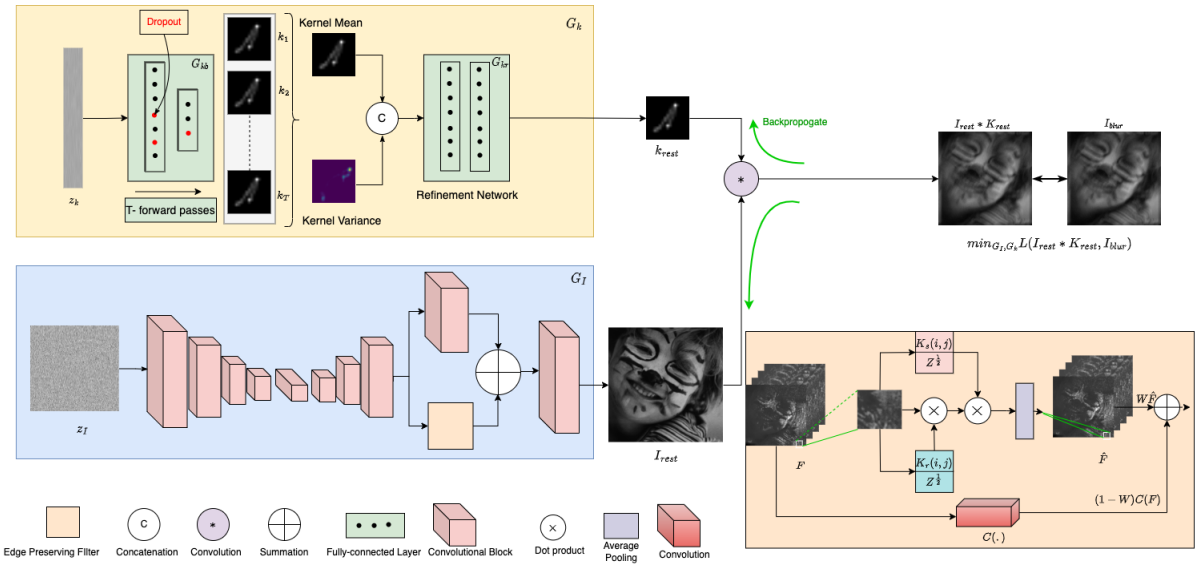

Fig. 2: An overview of the proposed network. Two generative networks networks G_I and G_K are used to generate the deep priors of the latent image. We use epistemic uncertainty of the base kernel generation network and estimate the mean and variance of the kernel which is then passed through a refinement network to get the final output kernel. The features of the penultimate layer is passed patch wise through the edge preserving filter.

diction network. These features are often close to the output of the network. Through our experiments we have found that by performing an artifact correction at the penultimate layer of the network, we can reduce ringing artifacts in the reconstructed image by a very good extent. In the case where the predicted output features of the image prior network has unwanted edges or artifacts, our network has the modelling capability to filter out the unwanted regions through an edge-preserving filter.

Our contributions are summarized as follows:

- We propose a new training strategy to generate better kernels in the double generative prior-based framework for blind image deconvolution.
- We propose a new gated edge preserving filter layer to improve the modeling of the image prior network.

2. PROPOSED METHOD

2.1. Network architecture

Double generative prior-based techniques have been proven effective for blind deblurring [12, 3]. Inspired by this, we develop a double generative prior-based network which consists of two main parts, the generator network for the blur kernel prior and the clean image prior. The overall pipeline of our network is illustrated in Fig.2. The novelties introduced are detailed as follows.

Kernel estimation network. Our kernel estimation network consists of two parts, the base kernel estimation network and the refinement module. Similar to SelfDeblur[3], we choose a two layered Fully connected Network (FCN) as our base kernel estimation network. But to make the network capable of estimating uncertainties in the network output, we add dropout to each layer with probability $p = 0.2$. While training our network, at each time step, we perform M number of forward passes with dropout, hence generating M number of kernels $k_1, ..., k_M$, we compute the mean (k_{mean}) and the variance (k_{var}) of the predicted kernels as

$$\begin{aligned} k_{mean} &= \frac{1}{M}\sum_{m=1}^{M} k_m \\ k_{var} &= \frac{1}{M}\sum_{m=1}^{M} k_m.k_m - k_{mean}.k_{mean}. \end{aligned} \tag{2}$$

The pixelwise variance measure of prediction k_{var} models the uncertainty in predicting the kernel. We develop a refinement module that utilizes this uncertainty and takes in k_{mean} and k_{var} as input. The architecture of the refinement module is a simple 2 layer FCN. As we know, most blur kernels are low pass in nature, hence to preserve this identity, the sum of all elements in the kernel matrix should be one. To ensure this, we add a softmax layer after the last layer of the base network and the refinement module.

Gated feature space edge preserving attention. To rectify artifacts caused due to small uncertainties in the kernel estimate, we include a new type of gated attention layer in our network. This layer has the capability of smoothing the effects of ringing artifacts and generating artifact-free features. Our attention mechanism is inspired from bilateral filtering[18]. Given the low-level features in the penultimate layer of the restoration network. Let these features be denoted by $F \in R^{(h\times w\times C)}$. For each channel of the low-level features, we do the following set of operations. Consider a feature channel $F_{h\times w}$ corresponding to a fixed c in F, a small local region of pixels is extracted around this pixel. Given a pixel x_a in the feature channel, we use a local region of pixels in positions $N_p(a)$ with spatial extent p centered around x_a

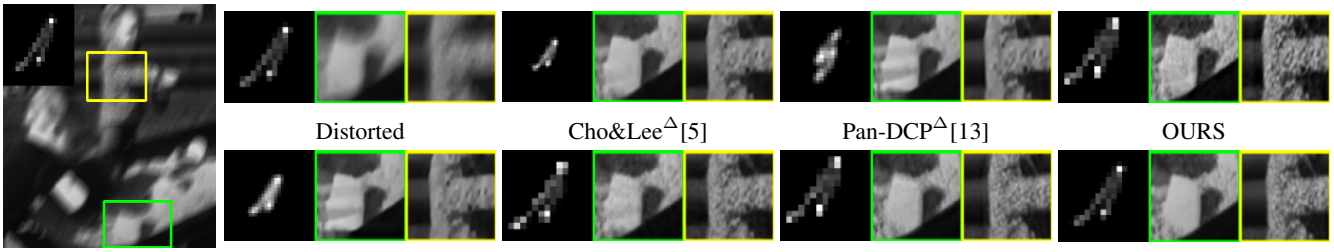

Fig. 3: Qualitative evaluations on the Levin et al. dataset [6]

Images	Cho&Lee$^{\Delta}$[5]	Xu&Jia$^{\Delta}$[15]	Sun$^{\Delta}$[16]	Zuo$^{\Delta}$[17]	Pan-DCP$^{\Delta}$[13]	SelfDeblur[3]	SelfDeblur$^{\Delta}$[3]	Ours	$Ours^{\Delta}$
PSNR	30.57	31.67	32.99	32.66	32.69	33.07	33.32	**33.65**	**33.87**
SSIM	0.896	0.916	0.933	0.933	0.928	0.931	0.943	**0.947**	**0.952**

Table 1: Average PSNR/SSIM comparison on the dataset of Levin et al.[6]. The methods marked with Δ uses existing non blind deblurring techniques for restoration.

to create two handcrafted kernels, the first kernel based on the intensity variation across this window and the other based on the spatial orientation of these pixels. Let $K_s(a,b)$ denote the spatial intensity kernel variation around a pixel location in consideration a and $b \in N_p(a)$ denote all the pixels in the neighbourhood of a. F_a and F_b denotes the intensity values at locations a and b, σ_r and σ_s are trainable parameters. We formally define these kernels as

$$K_s(a,b) = \frac{1}{Z^{\frac{1}{2}}} e^{\frac{-\|a-b\|^2}{2\sigma_s^2}}, K_r(a,b) = \frac{1}{Z^{\frac{1}{2}}} e^{\frac{-\|F_a-F_b\|^2}{2\sigma_r^2}}, \quad Z = \sum_{b\in N_p(a)} e^{\frac{-\|a-b\|^2}{2\sigma_s^2} + \frac{-\|F_a-F_b\|^2}{2\sigma_r^2}}. \tag{3}$$

The weights for each location are based on the intensity variation in the neighborhood of a. For $K_r(a,b)$, the weights for each location are based on the relative distance between the pixels. This could also be thought as a positional embedding kernel. Once these kernel functions are generated, the attended feature at location a can be estimated by

$$\hat{F_a} = \sum_{b\in N_p(a)} K_s(a,b)K_r(a,b)F(b) = \sum_{b\in N_p(a)} Softmax(H(F_a,F_b) + r(a,b))F(b), \tag{4}$$

where $H(.)$ is a function denoting the closeness of intensity values and $r(.)$ is the term denoting relative positions. The advantage of designing handcrafted attention rather than utilizing a standard attention module in this scenario is that the number of trainable parameters is just two. Also, this attention is computationally inexpensive and could be computed with linear complexity to the size of the image. Fig.2 shows the gated edge-preserving module. The inputs to the module are $\hat{F}$, which denotes the features after attention, and $C(F(.))$ denoting the features after a normal convolutional operation. The output of the gated network is,

$$O = W_1\hat{F} + (1 - W_1)C(F), \tag{5}$$

where the weights W_1 are computed by using a gate estimation network that takes as input the features F and the output of the restoration network during the previous m iterations.

Algorithm 1: Pseudo code for training our network.

Input: Blurry image **y**
Output: Clean image **x**, blur kernel **k**

```
for n = 1 : N do
    Sample z_k and z_x from uniform distribution.;
    for m = 1 : M do
        perform M forward passes;
        k_i = G_kb^{n-1}(z_k);
    end
    Find k_mean and k_var k_mean = (1/M) Σ_{m=1}^{M} k_m;
    k_var = (1/M) Σ_{m=1}^{M} k_m^T k_m − k_mean^T k_mean;
    k^n = G_kr^{n-1}(k_mean, k_var);
    x^n = G_x^{n-1}(z_x, {x^{n-1}, x^{n-2}, ..., x^{n-m+1}});
    Find gradients of G_kr, G_kb and G_x;
    Update parameters of G_kr, G_kb and G_x;
end
x = G_x^N(z_x, {x^{n-1}, x^{n-2}, ..., x^{n-m+1}});
```

The key idea is that artifacts present in the output during each iteration could be different. Algorithm 1 shows the pseudo code for the overall training process.

2.2. Loss functions

Let y denote the blurry image, x denote the clean latent image, $k_1, k_2, ..., k_M$ the kernel estimates found at a particular instant and k denote the output of the kernel refinement network. The net loss function for training our network is,

$$L = L_{MAP} + L_{mont} + \lambda L_{tv}, \tag{6}$$

where L denotes the net loss function for training our network. L_{MAP} focuses on finding the best parameters for the generative networks natural image estimation and kernel estimation that maximizes the posterior distribution $P(y|k,x)$. L_{tv} denotes the total variation loss for improving the quality of the predicted output x. Let $'h'$ and $'w'$ denote the axis along the length and width of the image. L_{mont} ensures

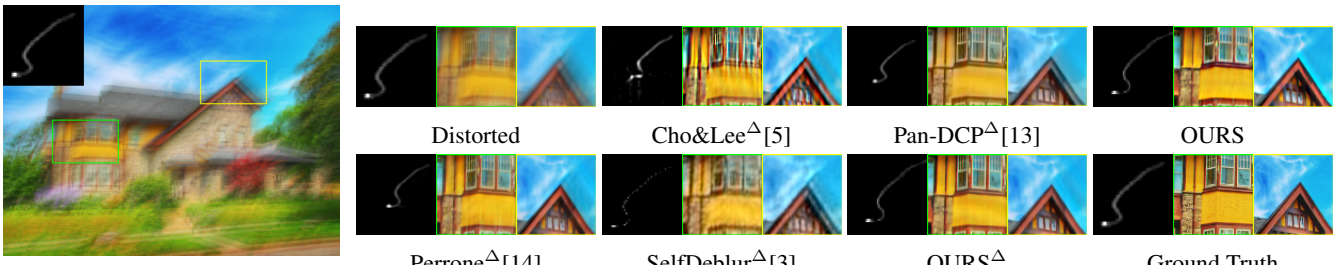

Fig. 4: Qualitative evaluations on the Lai et al. dataset [19]

Images	Xu&Jia$^{\Delta}$[15]	Xu$^{\Delta}$ [20]	Perroe et al.$^{\Delta}$[14]	Pan-DCP$^{\Delta}$[13]	$Self Deblur$[3]	$Self Deblur^{\Delta}$[3]	$Ours$	$Ours^{\Delta}$
$Manmade$	19.23/0.654	17.99/0.598	17.41/0.550	18.59/0.594	20.35/0.754	20.08/0.733	20.95/0.791	20.70/0.769
$Natural$	23.03/0.754	21.58/0.678	21.04/0.676	22.60/0.698	22.05/0.709	22.50/0.718	22.45/0.712	22.84/0.723
$People$	25.32/0.851	24.40/0.813	22.77/0.734	24.03/0.771	25.94/0.883	27.41/0.878	26.43/0.901	26.90/0.891
$Saturated$	14.79/0.563	14.53/0.538	14.24/0.510	16.57/0.632	16.35/0.636	16.58/0.616	16.61/0.665	16.91/0.633
$Text$	18.56/0.717	17.64/0.667	16.94/0.592	17.42/0.619	20.16/0.778	19.06/0.712	20.60/0.820	19.73/0.735
Avg	20.18/0.708	19.23/0.659	18.48/0.613	19.89/0.665	20.97/0.752	21.13/0.731	21.41/0.777	21.40/0.750

Table 2: Average PSNR/SSIM comparison on the dataset of Lai et al[19]

that the base kernel estimation network learns meaningful kernel representations and also the epistemic uncertainty of predicted kernels is minimized. k_j^{var} denotes each entry in the variance matrix of the kernel values $k_1, k_2, ..., k_M$. L_{MAP} L_{tv} and L_{mont} are defined as,

$$L_{MAP} = |x * k - y|^2, L_{tv} = \sqrt{|\frac{\partial x}{\partial h}|^2 + |\frac{\partial x}{\partial w}|^2},$$
$$L_{mont} = \frac{1}{2M}\left(\sum_{i=1}^{M}|x * k_i - y|^2 + \sum_{j \in k_{var}}|k_j^{var}|^2\right). \quad (7)$$

3. EXPERIMENTS

Training and Implementation details: Since our network uses the DIP framework, there is no explicit training of our network with data. The hyperparameters for the network are defined as follows. The number of iterations $N = 5000$. The number of forward passes through the base kernel estimation network $M = 10$. The value of the hyperparameter $\lambda = 10^{-7}$.The network is trained with Adam optimizer. The initial learning rate is set as 0.01. The learning rate is decayed by half after every 1000 iterations.

Evaluation benchmarks: We evaluate our network on the popular benchmark datasets from Levin *et al.*[6] and Lai *et al.*[19]. For comparisons, similar to [3, 16, 17], we use existing classical methods[5, 15, 16, 17, 13, 21, 14] for blind deblurring by estimating the blur kernels and then using the estimated blur kernels on the non-blind deblurring method by[22] to get the restored image. We also compare our method with the existing state-of-the-art deep learning technique, SelfDeblur[3]. For all comparison methods, we use the qualitative and quantitative results released by the authors of [3] and [19]. In the results shown, Δ denotes that the method from [6] has been used for evaluation using the estimated blur kernels. For evaluating the restored images, we utilize PSNR and SSIM as the metrics.

3.1. Results on the Levin *et al.*[6] and Lai *et al.*[19] dataset

The qualitative results on the Levin et al. dataset [6] are given in Fig.3. From the figure, we can see that the existing classical methods [16, 17, 13] over smooth the image. The method from [3] creates artifacts. In contrast, our method can preserve details without generating artifacts. Also, we can see from Table 1 that our method significantly outperforms all other methods in terms of PSNR and SSIM. The qualitative results on the Lai et al. dataset [19] are given in Fig.4. From the figure, we can see that recent deep learning-based technique [3] is unable to preserve fine details (such as windows) but generates significant artifacts for other regions. Methods such as [5, 14] fail to preserve details in the restored image. In contrast, our method is able to restore fine details as well as reduce artifacts to a good extent. Qualitative results show that our method is able to produce state-of-the-art results for most of the classes in terms of PSNR and SSIM. On average, our method performs better than all the existing methods. Table 3 shows the performance improvement from different modules proposed in our network.

Network	SelfDeblur[3]	SelfDeblur[3]+Image	SelfDeblur[3]+ Kernel	OURS
PSNR	33.07	33.31	33.50	33.65
SSIM	0.931	0.937	0.939	0.947

Table 3: Ablation study corresponding for the different modules in our network on Levin et al dataset[6]

4. CONCLUSION

We proposed an improved method for kernel estimation for blind image deblurring. We utilized the principle of epistemic uncertainty to predict the uncertainty in the prediction of kernels, and used this information to estimate better kernels. To account for the artifact problem in the restored image, we developed a feature space edge-preserving attention module. This attention module is applied to high-level features of the image restoration network to improve the quality of the restored image. Experiments performed on two benchmark datasets for blind deblurring showed that is able to estimate better blur kernels.

5. REFERENCES

[1] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas, "Deblurgan: Blind

motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8183–8192.

[2] Samuel Dodge and Lina Karam, "Understanding how image quality affects deep neural networks," in *2016 eighth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2016, pp. 1–6.

[3] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo, "Neural blind deconvolution using deep priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3341–3350.

[4] Ayan Chakrabarti, "A neural approach to blind motion deblurring," in *European conference on computer vision*. Springer, 2016, pp. 221–235.

[5] Sunghyun Cho and Seungyong Lee, "Fast motion deblurring," in *ACM SIGGRAPH Asia 2009 papers*, pp. 1–8. 2009.

[6] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman, "Understanding and evaluating blind deconvolution algorithms," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1964–1971.

[7] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454.

[8] Subeesh Vasu, Venkatesh Reddy Maligireddy, and AN Rajagopalan, "Non-blind deblurring: Handling kernel uncertainty with cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3272–3281.

[9] Yuesong Nan and Hui Ji, "Deep learning for handling kernel/model uncertainty in image deconvolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2388–2397.

[10] Alex Kendall and Yarin Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *arXiv preprint arXiv:1703.04977*, 2017.

[11] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[12] Yosef Gandelsman, Assaf Shocher, and Michal Irani, ""' double-dip": Unsupervised image decomposition via coupled deep-image-priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11026–11035.

[13] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang, "Deblurring images via dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 10, pp. 2315–2328, 2017.

[14] Daniele Perrone and Paolo Favaro, "Total variation blind deconvolution: The devil is in the details," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2909–2916.

[15] Li Xu and Jiaya Jia, "Two-phase kernel estimation for robust motion deblurring," in *European conference on computer vision*. Springer, 2010, pp. 157–170.

[16] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays, "Edge-based blur kernel estimation using patch priors," in *IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2013, pp. 1–8.

[17] Wangmeng Zuo, Dongwei Ren, David Zhang, Shuhang Gu, and Lei Zhang, "Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1751–1764, 2016.

[18] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens, "Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019.

[19] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang, "A comparative study for single image blind deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1701–1709.

[20] Li Xu, Shicheng Zheng, and Jiaya Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1107–1114.

[21] Tomer Michaeli and Michal Irani, "Blind deblurring using internal patch recurrence," in *European conference on computer vision*. Springer, 2014, pp. 783–798.

[22] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *CVPR 2011*. IEEE, 2011, pp. 2657–2664.