Single Image Defocus Deblurring via Implicit Neural Inverse Kernels

Yuhui Quan[1,2] Xin Yao[1,2] Hui Ji[3]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2] Pazhou Lab, Guangzhou 510335, China
[3] Department of Mathematics, National University of Singapore, 119076, Singapore
csyhquan@scut.edu.cn, csxinyaoscut@gmail.com, matjh@nus.edu.sg *

Abstract

Single image defocus deblurring (SIDD) is a challenging task due to the spatially-varying nature of defocus blur, characterized by per-pixel point spread functions (PSFs). Existing deep-learning-based methods for SIDD are limited by either over-fitting due to the lack of model constraints or under-parametrization that restricts their applicability to real-world images. To address the limitations, this paper proposes an interpretable approach that explicitly predicts inverse kernels with structural regularization. Motivated by the observation that defocus PSFs within an image often have similar shapes but different sizes, we represent the inverse kernels linearly over a multi-scale dictionary parameterized by implicit neural representations. We predict the corresponding representation coefficients via a duplex scale-recurrent neural network that jointly performs fine-to-coarse and coarse-to-fine estimations. Extensive experiments demonstrate that our approach achieves excellent performance using a lightweight model.

1. Introduction

When a camera captures an image, objects outside of the focal plane appear blurry. This effect is known as defocus blur and usually occurs when using a large camera aperture or capturing scenes with significantly varying depths. Removing defocus blur from a single image, known as SIDD, is desired for many practical applications; see *e.g.* [27, 14, 33, 60]. In general, defocus blur varies spatially. Each defocused pixel is a weighted average of its neighboring pixels in the latent all-in-focus image, with the weights determined by a spatially-varying unknown PSF.

SIDD is closely related to defocus map estimation (DME) [3, 44, 7, 5, 51, 32, 17]. Existing DME methods typically model defocus PSFs using a Gaussian function parameterized by its variance or a disc function parameterized by its radius. The resulting defocus map indicates the per-pixel blur amount in terms of the values of such model parameters, which can then be utilized for SIDD by applying a non-blind image deblurring (NID) algorithm using the corresponding PSFs; see *e.g.* [18]. However, DME remains a challenging task, and the estimated parameters of PSFs can be erroneous for many pixels. Moreover, Gaussian or disc-indicator functions are overly simplistic for real-world defocus PSFs. All these errors will be magnified in the NID process, resulting in noticeable artifacts.

Recently, many deep-learning-based methods have been proposed for SIDD; see *e.g.* [41, 25, 21, 47, 36, 40, 1, 56, 58, 23, 31]. Most train an end-to-end neural network (NN) that directly maps blurry images to their sharp correspondences. Although these methods provide better results than the DME-based two-step methods, their performance still has much room for improvement due to significant variations in the spatial distributions of defocus PSFs among different images. Many existing works (*e.g.* [21, 40, 56]) explicitly introduce spatially-varying processing blocks to better handle the spatial variance of defocus PSFs. However, those blocks lack constraints and may cause overfitting. A few studies (*e.g.* [36, 37]) represent defocus PSFs by some specific basis for geometric regularization and employ deep unrolling for an NN-based inversion process. Nevertheless, there is still room for improvement.

This paper presents an end-to-end deep NN with high interpretability for SIDD, which explicitly predicts spatially-varying inverse kernels of defocus PSFs and performs deblurring using the predicted inverse kernels. The possible overfitting issue is tackled by providing a more accurate model with structural constraints/regularization for the inverse kernels. Specifically, the inverse kernels are modeled by a linear representation under a dictionary. To achieve a compact yet sufficiently general representation, we construct the dictionary with a multi-scale structure. This is motivated by the empirical observation (as seen in [47]) that defocus PSFs within an image tend to have the same shape

*This work is supported by the Natural Science Foundation of Guangdong Province (Grant No. 2022A1515011755 and 2023A1515012841), and Singapore MOE AcRF Tier 1 Grant (WBS No. A-8000981-00-00).

but vary mainly in size. We show that if two defocus PSFs of different sizes have the same shape under an upsampling operation, so do the dictionary atoms of their inverse kernels. Thus, the dictionary atoms of a large size are defined as an upsampled version of the atoms of a small size.

Although plain upsampling offers an economical method for generating larger dictionary atoms from smaller ones, the resulting atoms may have a limited frequency range. This can lead to poor approximations of certain inverse kernels. To address the limitation, we leverage implicit neural representation (INR) [46], a technique that uses coordinate-input NNs to represent geometric objects like 2D shapes, so as to efficiently parameterize multi-scale atoms with both sufficient coverage of high-frequency content and implicit regularization to alleviate overfitting.

For the multi-scale representation coefficients, we separate them into scale-related and shape-related components, which are predicted by two scale-recurrent sub-NNs respectively. To better utilize information from different scales for coefficient prediction, we introduce a duplex scale-recurrent framework that performs both fine-to-coarse and coarse-to-fine estimations. Once the coefficients are predicted, the corresponding inverse kernels are applied to the input image for deblurring. The resulting pipeline is highly efficient and interpretable, leading to a lightweight yet effective model.

To conclude, this paper proposes an end-to-end NN for SIDD, which exhibits a noticeable performance boost compared to existing ones, while maintaining low complexity. See below for our main contributions:

- A parametric inverse kernel prediction framework for SIDD, providing efficient and interpretable deep NNs.
- A multi-scale linear representation model for the inverse kernels of spatially-varying defocus blur, using an INR-based multi-scale dictionary.
- A duplex scale-recurrent framework for predicting the representation coefficients of inverse kernels.

2. Related Works

2.1. Existing Approaches for SIDD

SIDD can done by a two-step approach that combines DME (*e.g*. [44, 51, 17, 32]) and NID (*e.g*. [20]). However, such an approach has two inherent drawbacks. First, the errors in DME will be magnified in the following NID process. Second, simple Gaussian or disc functions do not fit real-world defocus PSFs well. While there has been steady progress in deep-learning-based methods for DME [59, 16] and for NID [38, 10, 33, 29, 30, 4, 34, 35], the inherent drawbacks of the two-step approach remain.

Deep end-to-end learning has emerged as a promising approach for SIDD. The seminal work conducted by Abuolaim and Brown [2] trained an encoder-decoder CNN that maps a defocused image to its in-focus correspondence, with significant performance gain over two-step methods. To better handle spatially-varying defocus blur, some studies introduced dynamic processing blocks, *e.g*., per-pixel filter prediction in Lee *et al*. [21], dynamic residual blocks in Ruan *et al*. [40], and attention in Zhang and Zhai [56].

The use of dynamic processing blocks may increase the model complexity and the risk of overfitting. Regularization by auxiliary tasks is an effective approach for improving the generalization performance. In existing works, auxiliary tasks are often defined using DME ([25, 41, 58]), depth estimation [31], dual-view data [21, 1], or reblurring loss [23]. However, most of them require additional data sources.

A more appealing approach is encoding physical priors of defocus blur into NNs. Quan *et al*. [36, 37] used a kernel mixture model to parameterize defocus PSFs and constructed a deep unrolling NN. This method can be viewed as the degradation learning [22, 49, 24, 11] that predicts the blurring operator or kernels for deblurring. Our approach differs in that it predicts inverse kernels, not the degradation with defocus PSFs. Son *et al*. [47] showed that inverse defocus kernels have an invariant shape across scales and emulated them in feature spaces via a series of kernel-sharing parallel Atrous convolutions. Their method inspired our work, and it can be viewed as representing inverse kernels for a feature tensor via a multi-scale dilated combination of a single atom and predicting spatially-invariant combination coefficients via channel attention. In comparison, our approach explicitly predicts inverse kernels on the input image, allowing better interpretability, and represents them using INR to improve effectiveness. Additionally, we predict spatially-varying combination coefficients to better handle spatially-similar (not invariant) defocus shapes.

2.2. Related Techniques

Deep inverse kernels Several studies have included inverse kernels in deblurring NNs, in addition to [47]. Xu *et al*. [52] and Ren *et al*. [38] presented the CNNs that adopt tensor product for emulating separable inverse kernels to perform NID. Dong *et al*. [9, 10] and Pronina *et al*. [33] performed NID via Wiener inverse filtering in deep feature spaces. However, all these methods are designed for spatially-invariant blur. Ren *et al*. [55, 39] proposed a recurrent NN to mimic inverse kernels for removing spatially-varying blur of dynamic scenes, but not for SIDD.

Scale-recurrent NNs for deblurring Scale-recurrent NNs are commonly used for image deblurring, *e.g*. [36, 28, 48, 12, 54, 8, 15]. These methods typically employ coarse-to-fine estimation which is unidirectional and thus cannot fully leverage information from the subsequent scales for the prediction at the current scale. In contrast, our proposed NN utilizes a bi-directional structure via bi-LSTMs [13] for joint coarse-to-fine and fine-to-coarse estimation. This is

particularly useful for SIDD, as defocus blur effects show strong self-similarity over scales. While bi-LSTM has been used in [61] for multi-scale video deblurring, it is only used for bi-directional temporal processing and the multi-scale processing is still one-way. Also note that the multi-scale NN proposed in [6] for motion deblurring is not restrictive to unidirectional processing, but not in a recurrent manner.

INR for image recovery INR has been applied to image recovery [42, 19, 50]. In comparison to these works, we use INR to parameterize inverse kernels, not images, for SIDD.

3. INR-Based Dictionary for Inverse Kernels

3.1. Inverse Kernels for Spatially-Varying Blur

Consider a simplistic uniform blurring model defined by $\boldsymbol{y} = \boldsymbol{k} * \boldsymbol{x}$, where $*$ denotes convolution, $\boldsymbol{k}$ denotes the PSF, and $\boldsymbol{y}, \boldsymbol{x}$ denote the blurred image and its sharp correspondence, respectively. Let $\mathcal{F}$ denote the discrete Fourier transform (DFT). According to the well-known convolution theorem, we have $\mathcal{F}(\boldsymbol{y}) = \mathcal{F}(\boldsymbol{x}) \odot \mathcal{F}(\boldsymbol{k})$, where $\odot$ denotes entry-wise product. Then, one can define a (pesudo) inverse kernel $\boldsymbol{k}^\dagger$ by: for each (ω_x, ω_y),

$$\mathcal{F}(\boldsymbol{k}^\dagger)[\omega_x, \omega_y] = (\mathcal{F}(\boldsymbol{k})[s])^{-1}, \text{ if } |\mathcal{F}(\boldsymbol{k})[\omega_x, \omega_y]| \neq 0; \tag{1}$$

and 0 otherwise. Then, for an invertible PSF with only non-zero DFT coefficients, we have

$$\mathcal{F}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{k}^\dagger) \odot \mathcal{F}(\boldsymbol{y}) \Longrightarrow \boldsymbol{x} = \boldsymbol{k}^\dagger * \boldsymbol{y}. \tag{2}$$

When the PSF $\boldsymbol{k}$ is non-invertible, the result $\boldsymbol{k}^\dagger * \boldsymbol{y}$ still provides a fair approximation to $\boldsymbol{x}$.

One can extend (1) to handling spatially-varying blur. Let $\mathbb{K}^\dagger = \{\boldsymbol{k}^\dagger_{i,j}\}_{i,j}$ denote a set of inverse kernels. Then, the image $\boldsymbol{y}$ with spatially-varying blur can be recovered by a spatially-varying deconvolution process $\mathcal{C}$ defined by

$$\mathcal{C}\big(\boldsymbol{y}; \mathbb{K}^\dagger\big)[i, j] = \sum_{x,y} \boldsymbol{k}^\dagger_{i,j}[x, y]\boldsymbol{y}[i - x, j - y]. \tag{3}$$

Both the shape and the size of $\boldsymbol{k}^\dagger_{i,j}$ depend on the PSFs determined by scene depths and camera settings. When scene depths have significant variations, the sizes of the defocus PSFs also vary significantly, leading to large variations in the size of $\boldsymbol{k}^\dagger_{i,j}$ and a large number of unknowns in $\mathbb{K}^\dagger$. Without additional constraints on $\mathbb{K}^\dagger$, overfitting can occur.

3.2. Multi-Scale Dictionary for Inverse Kernels

To address the overfitting caused by excessive degrees of freedom in $\mathbb{K}^\dagger$, we parameterize the inverse kernels using a linear span over a dictionary with reduced number of atoms, resulting in inverse kernels that lie within a low-dimensional space. That is, we express an inverse kernel $\boldsymbol{k}^\dagger$ as a linear combination of atoms from a dictionary $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]$:

$$\boldsymbol{k}^\dagger = \omega_1 \boldsymbol{v}_1 + \cdots + \omega_N \boldsymbol{v}_N, \tag{4}$$

where $\omega_1, \cdots, \omega_N \in \mathbb{R}$ are the weights. Additionally, inverse kernels may have a large size. To accurately approximate them using as few atoms as possible, we introduce a multi-scale structure to the dictionary atoms, which is motivated by the observation that when two kernels have the same shape but different sizes, the larger one can be effectively approximated by upsampling the smaller one. Consequently, the dictionary atoms for the corresponding inverse kernels can also be related through such an upsampling process, as stated in the following proposition.

Proposition 1. *Let $\boldsymbol{k}^\dagger$ be an inverse kernel defined by (4). Let $\mathcal{U}_s$ be a standard dyadic upsampling operator with a factor s via expansion. Then*

$$\big(\mathcal{U}_s(\boldsymbol{k})\big)^\dagger = \omega_1 \mathcal{U}_s(\boldsymbol{v}_1) + \cdots + \omega_N \mathcal{U}_s(\boldsymbol{v}_N). \tag{5}$$

Proof. See supplemental material for the proof. □

Proposition 1 implies that two PFSs with the same shape but different sizes can share the same dictionary for their inverse kernels up to a spatial scaling factor. For defocus PSFs within an image, they typically have similar shapes related to camera settings, as observed in [47]. However, their sizes will vary with the corresponding scene depths. Thus, we represent the inverse kernels of PSFs using different upsampled versions of a dictionary with small atoms. Such a multi-scale structure provides a physics-based implicit prior on the PSFs within same image and leads to a more compact dictionary for representing inverse kernels of varying sizes. Specifically, we define a multi-scale dictionary composed by the sets (sub-dictionaries) $\{\boldsymbol{V}^r\}_{r=1}^R$:

$$\boldsymbol{V}^r = [\boldsymbol{v}^r_1, \cdots, \boldsymbol{v}^r_N] \subset \mathbb{R}^{M_r \times M_r}, \ \boldsymbol{v}^r_n = \mathcal{U}_{S_r}(\boldsymbol{v}_n), \tag{6}$$

where $\boldsymbol{v}^r_n$ denotes the atom at the r-th scale, S_r denotes an upsampling factor that determines the size parameter M_r of the atom. As r increases, S_r becomes larger and M_r gets smaller. The per-pixel inverse kernels are then expressed as

$$\boldsymbol{k}^\dagger_{i,j} = \omega_{1,i,j}\boldsymbol{v}_1^{r_{i,j}} + \cdots + \omega_{N,i,j}\boldsymbol{v}_N^{r_{i,j}} \in \mathbb{R}^{M_{r_{i,j}} \times M_{r_{i,j}}}, \tag{7}$$

where $r_{i,j}$ denotes the scale factor (index) related to $\boldsymbol{k}^\dagger_{i,j}$. Note that the atoms within the same sub-dictionary have a uniform size, while their sizes vary across different sub-dictionaries. That is, each sub-dictionary is utilized to represent inverse kernels of a specific size.

Using the linear representation (7) and the fact that $\boldsymbol{v}_n^{r_{i,j}} * \boldsymbol{y} = \sum_r \delta(r - r_{i,j})\boldsymbol{v}^r_n * \boldsymbol{y}$ with a Dirac delta function δ, we can rewrite Eq. (3) as

$$\mathcal{C}\big(\boldsymbol{y}\big) = \sum_{i,j} \mathbf{1}_{i,j} \odot \big(\sum_{n=1}^{N} \omega_{n,i,j}\boldsymbol{v}_n^{r_{i,j}} * \boldsymbol{y}\big) \tag{8}$$

$$= \sum_{i,j} \mathbf{1}_{i,j} \odot \big(\sum_{n=1}^{N}\sum_{r=1}^{R} \mu_{r,i,j}\omega_{n,i,j}\boldsymbol{v}^r_n * \boldsymbol{y}\big), \tag{9}$$

where $\mathbf{1}_{i,j}$ denotes a mask with 1 for the (i,j)-th entry being 1 and 0 otherwise, and $\mu_{r,i,j} = \delta(r - r_{i,j})$. We can then perform spatially-varying inverse filtering by convolving each atom $\boldsymbol{v}_n^r$ in the dictionary with the input image $\boldsymbol{y}$, and summing the resulting images by the spatially-varying weights $\mu_{s,i,j}\omega_{n,i,j}$. The weights $\mu_{s,i,j}$ and $\omega_{n,i,j}$ control the size (adaptive to scene depth) and shape (adaptive to image) of the corresponding inverse kernel, respectively.

3.3. INR Models for Dictionaries of Inverse Kernels

The dictionary $\{\boldsymbol{V}_r\}_{r=1}^R$ used in (8) is generated by the base atoms $\{\boldsymbol{v}_j\}_{j=1}^N$ of the smallest size. However, simply upsampling these base atoms cannot cover all high frequencies, *i.e.*, certain high-pass filters of large sizes may not be well approximated by the linear span of these atoms obtained by plain upsampling. To address this limitation, we use INR to re-parameterize the multi-scale dictionary. INR provides a more expressive way to generate multi-scale atoms that cover more frequencies than plain upsampling. Additionally, INR provides implicit regularization to alleviate overfitting by preferring image structures over random noise, as shown in [42, 19, 50].

Specifically, we express $\boldsymbol{v}_n^r$ as

$$\boldsymbol{v}_n^r[x,y] = \Phi_n(x,y),\ [x,y] \in [1,\cdots,M_r] \times [1,\cdots,M_r],$$

where $[x,y]$ denotes a spatial coordinate, and Φ is an INR model implemented by a compact multi-layer perceptron (MLP) of a small size. The INR model maps spatial coordinates to a continuous function that implicitly represents the kernel atom. This function can be evaluated at any point in space to interpolate a kernel atom of an arbitrary size, forming a dictionary of inverse kernels with arbitrary scales.

4. Implicit Neural Inverse Kernel Network

Our proposed NN called INIKNet consists of recurrent deblurring stages over different scales for exploiting cross-scale contextual information. Each deblurring stage predicts and applies inverse kernels to deblur the input image. Concretely, each stage consists of two steps: (i) given an input image $\boldsymbol{y}$, inferring two weight tensors $\boldsymbol{\mu}, \boldsymbol{\omega}$ that contain the weights $\{\mu_{r,i,j}\}_{r,i,j}$ and $\{\omega_{n,i,j}\}_{n,i,j}$ respectively; and (ii) applying inverse filtering defined by (8) to deblurring $\boldsymbol{y}$.

4.1. Duplex Scale-Recurrent Framework

Current scale-recurrent deblurring NNs, *e.g.* [36, 28, 48, 12, 54, 8, 15], typically use an LSTM-based coarse-to-fine framework. The deblurred image from the current (coarser) scale is attached to the input of the next (finer) scale, resulting in unidirectional feature flow that only considers coarser scales when inferring at finer scales. This approach cannot fully leverage the additional information available at finer scales for deblurring on coarser scales, especially when the blur effects show strong similarity over space and scale.

To address this limitation, we propose a duplex multi-scale processing framework, which leverages bi-LSTM [13] as the sub-NNs to predict $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$. It improves predictions at coarser scales by incorporating additional information from finer scales through the feature flow in the fine-to-coarse pass. This refined information is then utilized to further improve predictions on finer scales via the feature flow through the coarse-to-fine pass. As a result, the prediction at the current scale benefits from the information extracted from both the previous and subsequent scales.

As shown in Fig. 1, our INIKNet first downsamples the input image to multi-scale versions. For each scale, a U-Net shared across scales is first applied for extracting spatial features from the input. The extracted features are then fed to two bi-LSTM sub-NNs so as to exploit the dependencies among scales for feature refinement. The outputs of two bi-LSTMs sub-NNs are then used to predict $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$ via an 1×1 convolutional layer, respectively. Afterward, the predicted weights are used by an Inverse Filtering Module (IFM) to perform deblurring using (8) to output the result.

Our INIKNet differs from existing coarse-to-fine NNs in that it does not concatenate the output image from the current scale to the input of the next scale, as this operation would not only make the feature flow unidirectional, but also prevent the current-scale processing from starting, until the deblurring is fully completed at the previous scale. By removing this operation, the U-Net and bi-LSTM can start running earlier, without waiting for the deblurred images from previous scales. Moreover, by cancelling the operation, the IFM only needs to be called at the original image scale during inference, even though it is still required during training to calculate the multi-scale loss.

4.2. Details of Key Modules

Bi-LSTM sub-NNs Recall that the weights $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$ correspond to two distinct properties of inverse kernels, namely size and shape, respectively. Their dependency on image scales also differs significantly. Therefore, rather than use a single bi-LSTM sub-NN to predict $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$ jointly, we use two parallel bi-LSTM sub-NNs for separate predictions. Both sub-NNs share the same structure, consisting of convolutional LSTM cells [45] arranged in two paths for fine-to-coarse and coarse-to-fine estimations in opposite directions. Each LSTM cell takes a cell state and a hidden state as input and outputs updated states. The input cell state is defined as the features extracted by the U-Net, while the input hidden state for the next scale is set to the output one from the previous scale, forming a cross-scale connection.

Weight predictors To convert the output features from the bi-LSTM sub-NNs to the weights $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$, two 1×1 convolutional layers are applied, respectively. For predicting $\boldsymbol{\mu}$, the Softmax function is used to impose non-negativity and ℓ_1-normalization constraints. Such a soft relaxation of

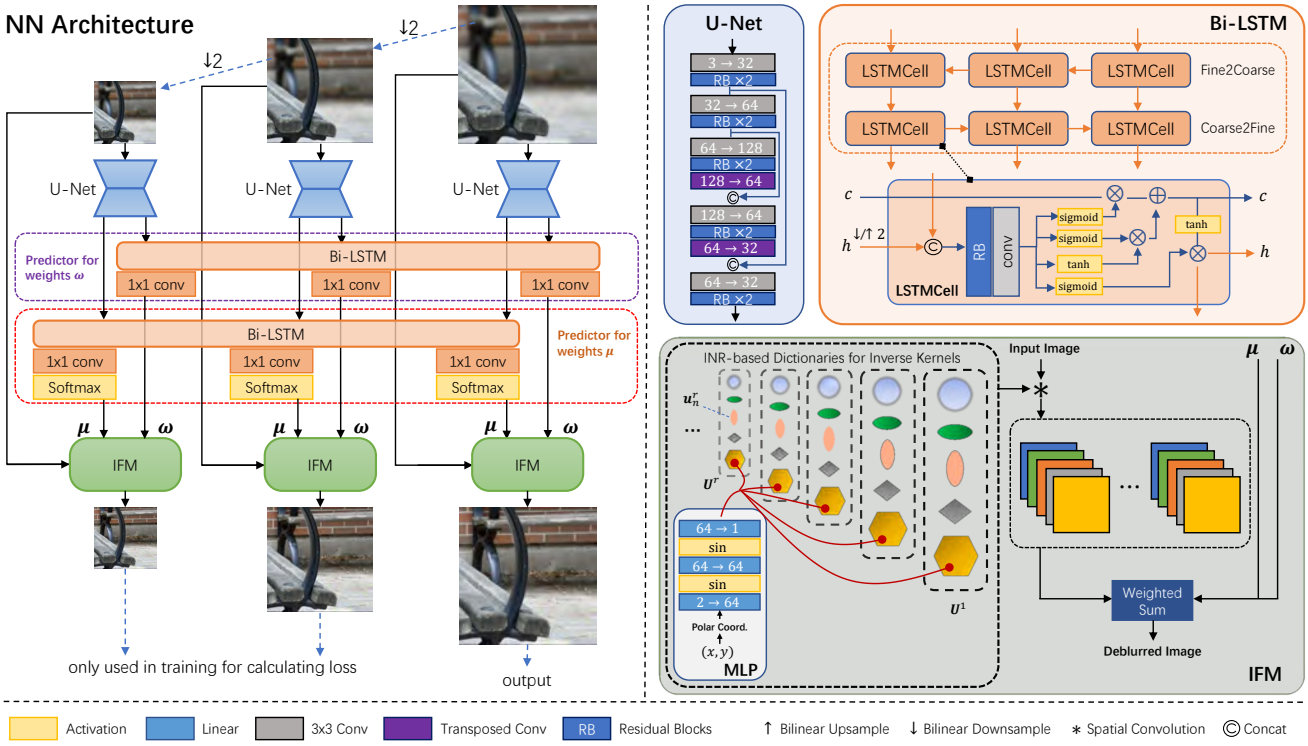


Figure 1: Architecture of proposed INIKNet for SIDD.

weights $\mu_{r,i,j}$ from $0, 1$ to $[0, 1]$ not only makes training easier, but also improves the representation accuracy of inverse kernels by including the use of additional atoms. This can be useful when the inverse kernel sizes fall between the gaps among the predefined non-consecutive atom sizes.

INR models of kernel atoms The MLPs used for INR have a structure similar to those in [46]. They comprise three fully-connected (FC) layers separated by a Sine activation function with a learnable frequency parameter that controls frequency content. The 2D coordinates are first mapped to a high-dimensional space and then projected to single values. One difference from [46] is that we first transform spatial coordinates to polar coordinates for slight improvement.

IFM for deblurring The IFM first calls the learned MLPs to generate all kernel atoms $\{\boldsymbol{v}_n^r\}_{n,r}$ and then calculates the feature tensors $\boldsymbol{f}[n, r] = \boldsymbol{v}_n^r * \boldsymbol{y}$. Afterward, Eq. (8) is implemented by simple entry-wise product and summation operations on the tensors $\boldsymbol{f}, \boldsymbol{\mu}, \boldsymbol{\omega}$.

4.3. Training Loss

In addition to the ℓ_2 loss $\mathcal{L}_{\text{MSE}}$ defined by mean squared error (MSE), two loss functions often seen in image recovery are also used for training, including the frequency-domain loss $\mathcal{L}_{\text{FD}}$ [6] defined as the ℓ_1 distance in the frequency domain, and the Learned Perceptual Image Patch Similarity (LPIPS) loss $\mathcal{L}_{\text{LPIPS}}$ [57] predicted by a pre-trained NN. These three loss functions measure deblurring quality in the spatial, frequency and feature (perceptual) domains, respectively. The total loss is then given by

$$\mathcal{L} = \sum_p \mathcal{L}_{\text{MSE}}^{(p)} + \lambda_1 \mathcal{L}_{\text{FD}}^{(p)} + \lambda_2 \mathcal{L}_{\text{LPIPS}}^{(p)}, \tag{10}$$

where p denotes the index of the pth scale, and λ_1, λ_2 are empirically set to 0.1. Ground-truth images are downsampled accordingly for the supervision at different scales.

5. Experiments

Datasets and metrics Following [2, 41], we train models on two widely-used datasets, DPDD [2] and LFDOF [41], respectively. To assess their generalization performance, we also evaluate these models on two more datasets, RTF [7] and RealDOF [21]. Three quality metrics are used for quantitative evaluation: Peak Signal to Noise Ratio (PSNR, measured in dB), Structural SIMilarity index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [57]. The model complexity is measured by both the number of parameters and the inference time on a 640×640 image using an NVIDIA 2080Ti RTX GPU.

Implementation details Throughout the experiments, we employ 8 scales and 10 INR-based atoms per scale: $R = 8$ and $N = 10$. We use atoms of sizes $1\times1, 3\times3, \cdots, 15\times15$,

where the 1×1 atom is fixed as a delta kernel for modeling in-focus regions. The frequency parameters of the sine activation functions in all INR models are initialized by random sampling from $[2, 16]$. We call Adam for training, with 4×10^5 iterations and a batch size of 4. We start with an initial learning rate of 2×10^{-4} and halve it every 10^5 iterations. Random flipping, rotation, and cropping (to 256×256 pixels) are performed for training data augmentation.

5.1. Performance Evaluation

Compared methods We select 9 NN-based SIDD methods for comparison, including the DPDNet [2], AIFNet [41], IFANet [21], KPAC [47], GKMNet [36], MDP [1], DRBNet [40], Restormer [53], and MPRNet [26] trained in [23] with a misalignment-robust scheme. The experimental results of the compared methods are reported either from their original papers or obtained by using their released pre-trained models and codes. AIFNet is a model trained on LFDOF with the additional use of SYNDOF for its DME step. To compare its performance on DPDD, we retrain it using both DPDD and SYNDOF. DRBNet is a model trained on LFDOF and fine-tuned on DPDD. For a fair comparison, we retrain it on both datasets, respectively.

Evaluation on DPDD and LFDOF Table 1 presents the quantitative results on DPDD and LFDOF. Our INIKNet performs the best in PSNR on DPDD and in all three metrics on LFDOF, despite using a lightweight model whose parameter number is the second smallest and around 13.2% of that of Restormer, a general restoration model. Compared to the latest SIDD-specified MRPNet model [23], INIKNet achieves over 0.3 dB PSNR improvement on DPDD while having only about 10% parameters to learn. Similar results are observed in the comparison with DRBNet, indicating that the performance gain of INIKNet comes from the design of NN structure rather than the increased model size. Though INIKNet is slightly larger than GKMNet, the smallest model, it achieves noticeably better performance. Compared to KPAC that uses dilated convolutions to mimic inverse kernels, INIKNet performs significantly better. In conclusion, the superiority and compactness of INIKNet demonstrate the effectiveness of its architecture design.

Generalization analysis on RealDOF and RTF The results of the DPDD/LFDOF-trained models evaluated on RealDOF and RTF are listed in Table 2. Our DPDD-trained INIKNet shows good generalization to different datasets and achieves the best performance on both RealDOF and RTF in all three metrics. The LFDOF-trained INIKNet is the top performer on RealDOF and achieves the best SSIM score on RTF, as well as the second-best scores in PSNR and LPIPS. Though AIFNet performs better in PSNR and LPIPS, it requires an additional dataset for DME and is much larger than INIKNet. In summary, INIKNet exhibits superior generalizability over other compared models.

Model	DPDD			LFDOF			#Params	Time
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	(10^6)	(sec.)
DPDNet	24.348	0.747	0.277	-	-	-	32.25	1.430
AIFNet	24.213	0.742	0.309	29.677	0.884	0.202	41.55	0.276
IFANet	25.366	0.789	0.217	29.787	0.872	0.156	10.48	0.063
KPAC	25.221	0.774	0.226	28.942	0.857	0.174	2.06	**0.037**
GKMNet	25.468	0.789	0.219	29.081	0.867	0.171	**1.41**	0.129
MDP	25.347	0.763	0.268	28.069	0.834	0.185	46.86	3.317
DRBNet	25.485	0.792	0.254	30.253	0.883	0.147	11.69	0.085
MPRNet	25.730	0.792	0.232	-	-	-	20.10	1.161
Restormer	25.980	**0.811**	**0.178**	-	-	-	26.10	0.672
INIKNet	**26.055**	0.803	0.185	**30.293**	**0.886**	**0.132**	1.98	0.228

Table 1: Quantitative results on DPDD/LFDOF. **Bold** for best performers and underline for second-best performers.

	Model	RealDOF			RTF			#Params
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	(10^6)
Trained on DPDD	DPDNet	22.870	0.670	0.425	23.608	0.591	0.296	32.25
	AIFNet	23.093	0.680	0.413	24.041	0.758	0.289	41.55
	MDP	23.500	0.681	0.444	24.012	0.738	0.312	46.86
	KPAC	23.975	0.762	0.338	24.618	0.777	0.236	2.06
	IFANet	24.712	0.748	0.306	24.924	0.801	0.227	10.48
	GKMNet	24.254	0.732	0.392	24.970	0.789	0.261	**1.41**
	DRBNet	24.884	0.751	0.376	24.463	0.773	0.311	11.69
	MPRNet	24.541	0.736	0.339	24.588	0.788	0.304	20.10
	Restormer	25.091	0.762	0.285	24.212	0.821	0.224	26.10
	INIKNet	**25.231**	**0.765**	**0.287**	**25.450**	**0.834**	**0.215**	1.98
Trained on LFDOF	IFANet	22.504	0.669	0.483	26.437	0.838	0.238	10.48
	KPAC	22.550	0.671	0.457	25.959	0.803	0.230	2.06
	AIFNet	22.623	0.667	0.461	**27.552**	0.882	**0.176**	41.55
	GKMNet	23.609	0.721	0.408	26.985	0.856	0.246	**1.41**
	MDP	22.726	0.680	0.453	25.580	0.809	0.228	46.86
	DRBNet	22.910	0.691	0.437	26.717	0.853	0.200	11.69
	INIKNet	**23.810**	**0.724**	**0.356**	27.401	**0.885**	0.179	1.98

Table 2: Quantitative results on RealDOF/RTF. **Bold** for best performers and underline for second-best performers.

Qualitative comparison We compare INIKNet with some SIDD-specified competitors on DPDD and LFDOF via visual inspection. See Fig. 2 for some deblurred images on four datasets, where complex textures such as those on wire fence and branches are successfully recovered by INIKNet. See also Fig. 3 for some results on the CUHK dataset [43]. While most methods fail under large blur, ours yields more visually satisfying results. These findings demonstrate that our INIKNet performs well in terms of visual quality.

5.2. Ablation Studies and Analysis

Ablation studies We construct several variants of INIKNet and list their results on two datasets in Table 3.

Non-MS Dict: We drop the multi-scale structure of the dictionary and instead use an individual INR model for each dictionary atom at each scale (thus yielding a larger model size). A noticeable performance degradation occurs. This is because the multi-scale structure of the INR-based dictionary imposes structural regularization to reduce overfitting.

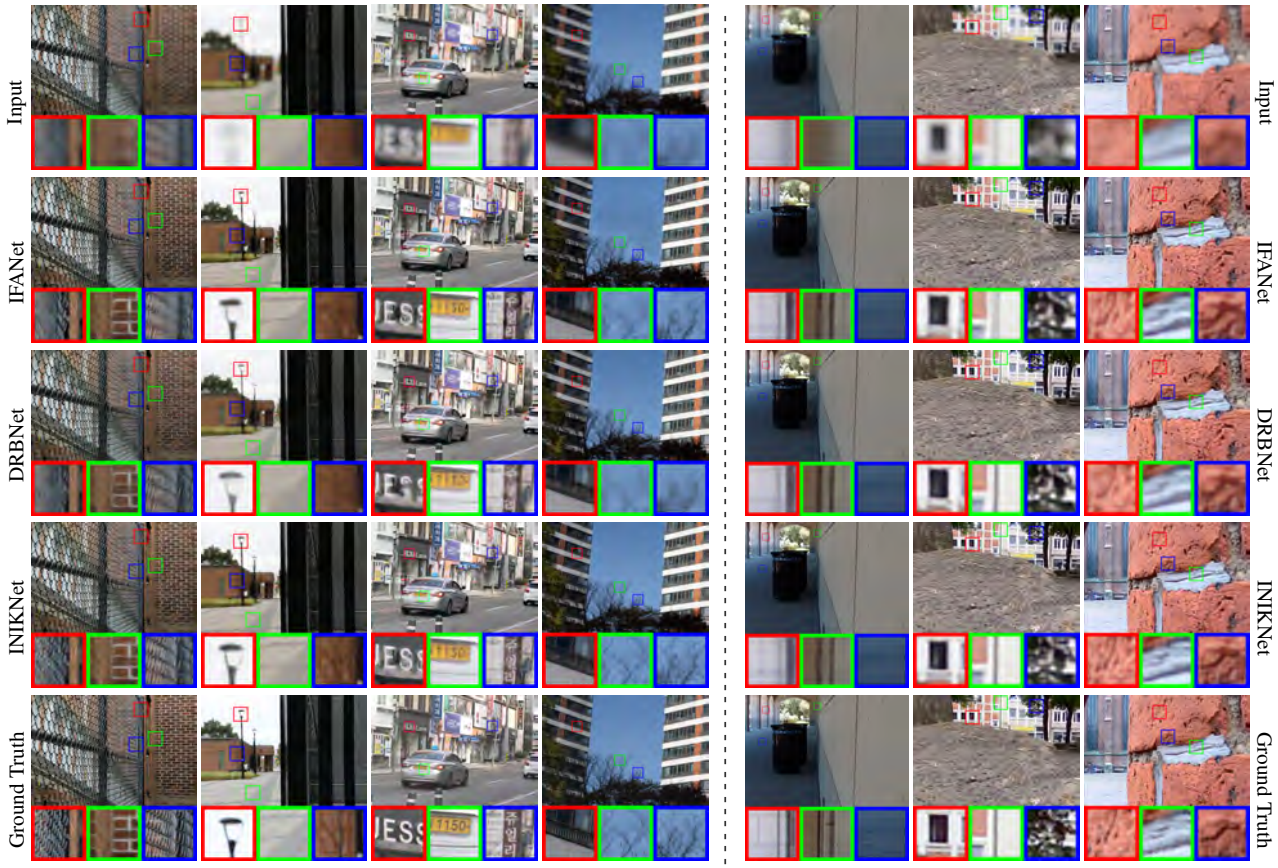


Figure 2: Results on using models trained on DPDD (left part) and LFDOF (right part). Zoom-in for better views.

Model	RealDOF			RTF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Non-MS Dict	24.916	0.749	0.310	25.136	0.809	0.254
Non-INR Dict	25.058	0.754	0.303	25.237	0.819	0.226
Unidirectional	25.099	0.753	0.294	25.305	0.824	0.227
Original INIKNet	25.231	0.765	0.287	25.450	0.834	0.215

Table 3: Results of baseline models trained on DPDD.

Non-INR Dict: When the INR-based dictionaries are replaced by the plain ones, where each base atom $\boldsymbol{v}_n$ is defined as a learnable matrix and bi-linearly upsampled to form the multi-scale dictionary, the performance of INIKNet decreases. This is probably because the frequency content for the atoms generated by plain upsampling are limited, so do their generated inverse kernels.

Unidirectional: To simulate an unidirectional scale recurrent mechanism (*e.g.* [36]), we replace the bi-LSTM with double unidirectional LSTMs (for keeping the model size). The output from the last scale is attached to the input of the current scale for additional information. Such a scheme for coefficient prediction results in some performance degradation, highlighting the effectiveness of our proposed duplex scale-recurrent framework.

Visualization of coefficients and atoms The coefficient maps $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$ are visualized in Fig. 4. We can see that the regions with larger blur amount tend to have larger coefficients $\boldsymbol{\mu}$ assigned to larger inverse kernels, and vice versa. In comparison, the coefficients $\boldsymbol{\omega}$ are more spatially similar (gray), consistent with the prior of spatially-similar shapes of inverse kernels. We use Fig. 5 to visualize some learned (non-)INR-based atoms in both the spatial and Fourier domain. The INR-based atoms within the same scale contain diverse patterns with varying frequencies and orientations, and meanwhile they have similar shapes across scales. Compared to the non-INR-based ones, the INR-based atoms encode richer spatial structures and frequency contents. Interestingly, the INR-based atoms exhibit strong isotropy, which is likely due to the isotropy of defocus PSFs (so do their inverse kernels). All these properties of the INR-based atoms enable our INIKNet to effectively represent and predict inverse kernels for spatially-varying defocus blur.

6. Conclusion

This paper presented an interpretable end-to-end SIDD approach, which learns a duplex scale-recurrent NN to predict INR-driven dictionary-parameterized inverse kernels of

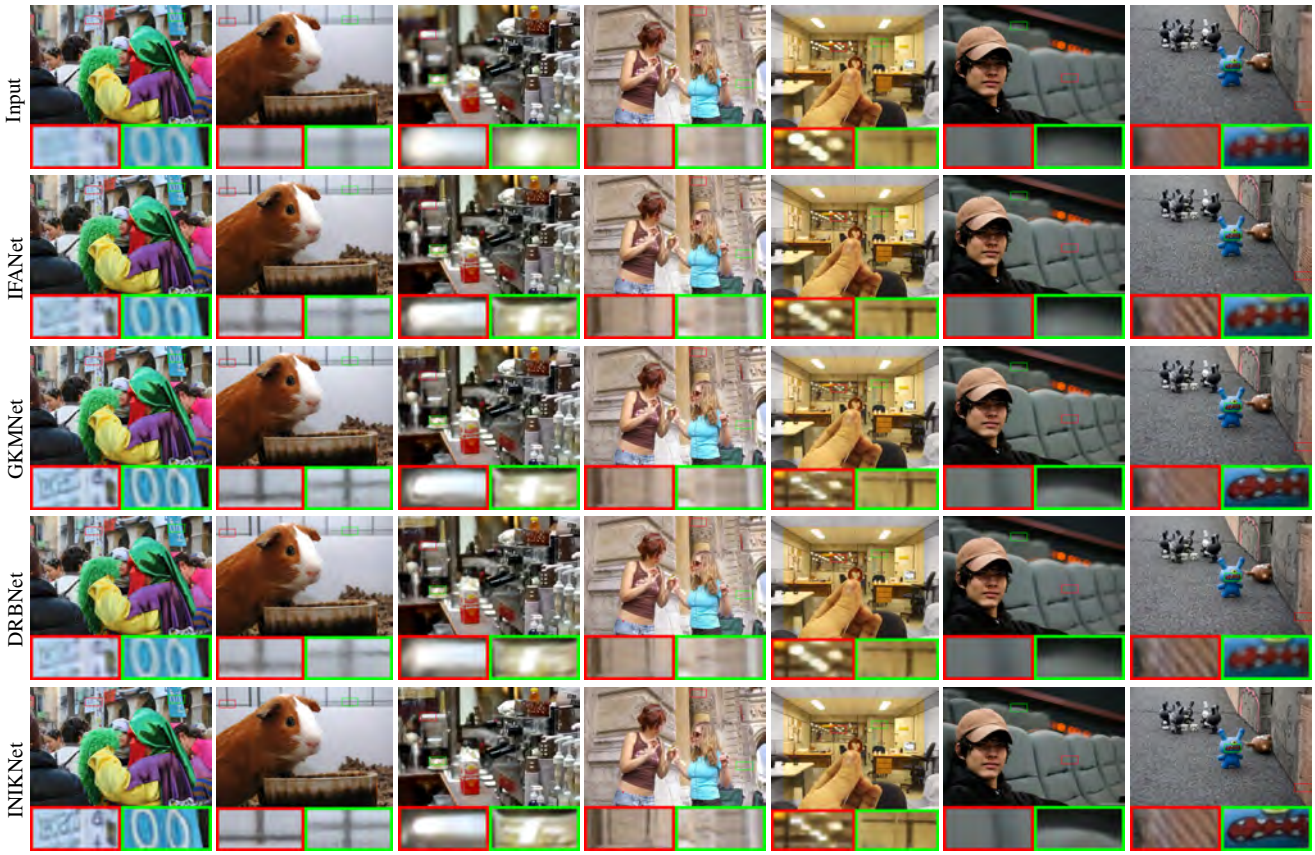


Figure 3: Results on CUHK dataset (without ground-truths) using models trained on DPD. Zoom-in for better views.

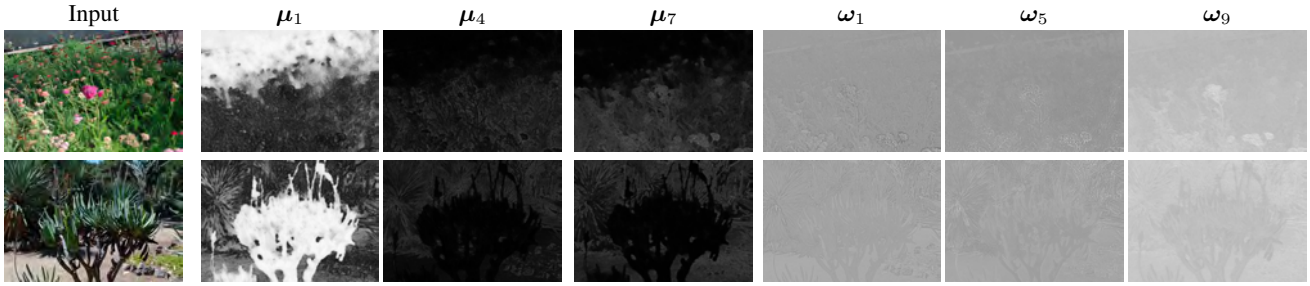


Figure 4: Visualization of μ maps and ω maps.

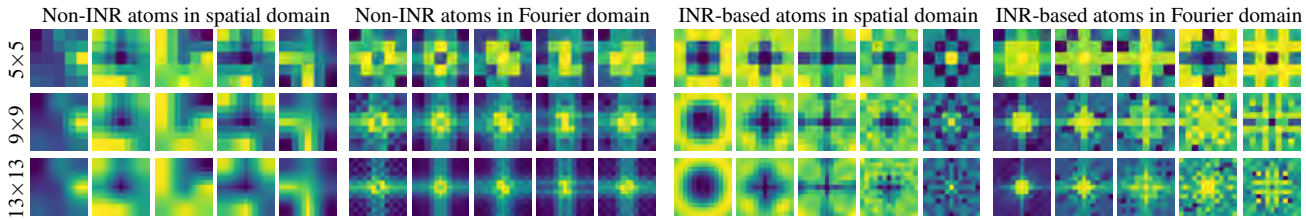


Figure 5: Visualization of bilinearly upsampled (non-INR-based) atoms and INR-based atoms in two domains.

defocus blur and applies them to deblur images. Our proposed approach introduced effective structural regularization on inverse kernels for good generalization, which is based on the shape similarity of defocus PSFs over an image. The resulting effective and light-weight model has demonstrated its advantages by extensive experiments.

References

[1] Abdullah Abuolaim, Mahmoud Afifi, and Michael S. Brown. Improving single-image defocus deblurring: How dual-pixel images help through multi-task learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1231–1239, 2022. 1, 2, 6

[2] Abdullah Abuolaim and Michael S. Brown. Defocus deblurring using dual-pixel data. In *Proceedings of the European Conference on Computer Vision*, pages 111–126. Springer, 2020. 2, 5, 6

[3] Yosuke Bando and Tomoyuki Nishita. Towards digital refocusing from a single photograph. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*, pages 363–372. IEEE, 2007. 1

[4] Mingqin Chen, Yuhui Quan, Tongyao Pang, and Hui Ji. Nonblind image deconvolution via leveraging model uncertainty in an untrained deep neural network. *International Journal of Computer Vision*, 130(7):1770–1789, 2022. 2

[5] Sunghyun Cho and Seungyong Lee. Convergence analysis of map based blur kernel estimation. In *Proceedings of the IEEE/International Conference on Computer Vision*, pages 4808–4816, 2017. 1

[6] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4641–4650, 2021. 3, 5

[7] Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. Non-parametric blur map regression for depth of field extension. *IEEE Transactions on Image Processing*, 25(4):1660–1673, 2016. 1, 5

[8] Senyou Deng, Wenqi Ren, Yanyang Yan, Tao Wang, Fenglong Song, and Xiaochun Cao. Multi-scale separable network for ultra-high-definition video deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14030–14039, 2021. 2, 4

[9] Jiangxin Dong, Stefan Roth, and Bernt Schiele. Deep wiener deconvolution: Wiener meets deep learning for image deblurring. *Advances in Neural Information Processing Systems*, 33:1048–1059, 2020. 2

[10] Jiangxin Dong, Stefan Roth, and Bernt Schiele. Dwdn: deep wiener deconvolution network for non-blind image deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9960–9976, 2021. 2

[11] Zhenxuan Fang, Fangfang Wu, Weisheng Dong, Xin Li, Jinjian Wu, and Guangming Shi. Self-supervised non-uniform kernel estimation with flow-based motion prior for blind image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18105–18114, 2023. 2

[12] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3848–3856, 2019. 2, 4

[13] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005. 2, 4

[14] Fang Hua, Peter Johnson, Nadezhda Sazonova, Paulo Lopez-Meyer, and Stephanie Schuckers. Impact of out-of-focus blur on face recognition performance based on modular transfer function. In *Proceedings of the International Conference on Biometrics*, pages 85–90. IEEE, 2012. 1

[15] Bo Ji and Angela Yao. Multi-scale memory-based video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1919–1928, 2022. 2, 4

[16] Ali Karaali, Naomi Harte, and Claudio R. Jung. Deep multiscale feature learning for defocus blur estimation. *IEEE Transactions on Image Processing*, 31:1097–1106, 2022. 2

[17] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Transactions on Image Processing*, 27(3):1126–1137, Mar. 2018. 1, 2

[18] Ali Karaali and Cláudio Rosito Jung. Svbr-net: A non-blind spatially varying defocus blur removal network. In *Proceedings of the IEEE International Conference on Image Processing*, pages 566–570. IEEE, 2022. 1

[19] Chaewon Kim, Jaeho Lee, and Jinwoo Shin. Zero-shot blind image denoising via implicit neural representations. *arXiv preprint arXiv:2204.02405*, 2022. 3, 4

[20] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, 2009. 2

[21] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2034–2042, 2021. 1, 2, 6

[22] Dasong Li, Yi Zhang, Ka Chun Cheung, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Learning degradation representations for image deblurring. In *Proceedings of the European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2

[23] Yu Li, Dongwei Ren, Xinya Shu, and Wangmeng Zuo. Learning single image defocus deblurring with misaligned training pairs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 1, 2, 6

[24] Ziwei Luo, Haibin Huang, Lei Yu, Youwei Li, Haoqiang Fan, and Shuaicheng Liu. Deep constrained least squares for blind image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17642–17652, 2022. 2

[25] Haoyu Ma, Shaojun Liu, Qingmin Liao, Juncheng Zhang, and Jing-Hao Xue. Defocus image deblurring network with defocus map estimation as auxiliary task. *IEEE Transactions on Image Processing*, 31:216–226, 2021. 1, 2

[26] Armin Mehri, Parichehr B Ardakani, and Angel D Sappa. Mprnet: Multi-path residual network for lightweight image super resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2704–2713, 2021. 6

[27] Sung-Jun Min, Kyeongbo Kong, and Suk-Ju Kang. Out-of-focus image deblurring for mobile display vision inspection.

IEEE Transactions on Circuits and Systems for Video Technology, 2023. 1

[28] Seungjun Nah, Tae Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 257–265, 2017. 2, 4

[29] 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*, pages 2388–2397, 2020. 2

[30] Yuesong Nan, Yuhui Quan, and Hui Ji. Variational-EM-based deep learning for noise-blind image deblurring. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3626–3635, 2020. 2

[31] Saqib Nazir, Lorenzo Vaquero, Manuel Mucientes, Víctor M Brea, and Daniela Coltuc. Depth estimation and image restoration by deep learning from defocused images. *arXiv preprint arXiv:2302.10730*, 2023. 1, 2

[32] Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and In So Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1736–1745, 2017. 1, 2

[33] Valeriya Pronina, Filippos Kokkinos, Dmitry V Dylov, and Stamatios Lefkimmiatis. Microscopy image restoration with deep wiener-kolmogorov filters. In *Proceedings of the European Conference on Computer VIsion*, pages 185–201. Springer, 2020. 1, 2

[34] Yuhui Quan, Zhuojie Chen, Huan Zheng, and Hui Ji. Learning deep non-blind image deconvolution without ground truths. In *Proceedings of the European Conference on Computer Vision*, pages 642–659, 2022. 2

[35] Yuhui Quan, Peikang Lin, Yong Xu, Yuesong Nan, and Hui Ji. Nonblind image deblurring via deep learning in complex field. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5387–5400, 2022. 2

[36] Yuhui Quan, Hui Wu, Zicong, and Hui Ji. Gaussian kernel mixture network for single image defocus deblurring. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 1, 2, 4, 6, 7

[37] Yuhui Quan, Hui Wu, Zicong, and Hui Ji. Neumann network with recursive kernels for single image defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5754–5763, June 2023. 1, 2

[38] Wenqi Ren, Jiawei Zhang, Lin Ma, Jinshan Pan, Xiaochun Cao, Wangmeng Zuo, Wei Liu, and Ming-Hsuan Yang. Deep non-blind deconvolution via generalized low-rank approximation. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[39] Wenqi Ren, Jiawei Zhang, Jinshan Pan, Sifei Liu, Jimmy S Ren, Junping Du, Xiaochun Cao, and Ming-Hsuan Yang. Deblurring dynamic scenes via spatially varying recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):3974–3987, 2021. 2

[40] Lingyan Ruan, Bin Chen, Jizhou Li, and Miuling Lam. Learning to deblur using light field generated and real defocus images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16304–16313, 2022. 1, 2, 6

[41] Lingyan Ruan, Bin Chen, Jizhou Li, and Miu-Ling Lam. Aifnet: All-in-focus image restoration network using a light field-based dataset. *IEEE Transactions on Computational Imaging*, 7:675–688, 2021. 1, 2, 5, 6

[42] Nimrod Shabtay, Eli Schwartz, and Raja Giryes. Pip: Positional-encoding image prior. *arXiv preprint arXiv:2211.14298*, 2022. 3, 4

[43] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2972, 2014. 6

[44] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–665, 2015. 1, 2

[45] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28, 2015. 4

[46] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 2, 5

[47] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2642–2650, 2021. 1, 2, 3, 6

[48] Xin Tao, Hongyun Gao, Yi Wang, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 2, 4

[49] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10581–10590, 2021. 2

[50] Shaowen Xie, Hao Zhu, Zhen Liu, Qi Zhang, You Zhou, Xun Cao, and Zhan Ma. Diner: Disorder-invariant implicit neural representation. *arXiv preprint arXiv:2211.07871*, 2022. 3, 4

[51] Guodong Xu, Yuhui Quan, and Hui Ji. Estimating defocus blur via rank of local patches. In *Proceedings of the IEEE/International Conference on Computer Vision*, pages 5371–5379, 2017. 1, 2

[52] Li Xu, Jimmy S Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. *Advances in Neural Information Processing Systems*, 27, 2014. 2

[53] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang.

Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 6

[54] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5978–5986, 2019. 2, 4

[55] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2521–2529, 2018. 2

[56] Jie Zhang and Wanming Zhai. Blind attention geometric restraint neural network for single image dynamic/defocus deblurring. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1, 2

[57] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5, 6

[58] Wenda Zhao, Fei Wei, You He, and Huchuan Lu. United defocus blur detection and deblurring via adversarial promoting learning. In *Proceedings of the European Conference on Computer Vision*, pages 569–586, 2022. 1, 2

[59] Wenda Zhao, Fan Zhao, Dong Wang, and Huchuan Lu. Defocus blur detection via multi-stream bottom-top-bottom network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:1884–1897, 2020. 2

[60] Bolun Zheng, Quan Chen, Shanxin Yuan, Xiaofei Zhou, Hua Zhang, Jiyong Zhang, Chenggang Yan, and Gregory Slabaugh. Constrained predictive filters for single image bokeh rendering. *IEEE Transactions on Computational Imaging*, 8:346–357, 2022. 1

[61] Chao Zhu, Hang Dong, Jinshan Pan, Boyang Liang, Yuhao Huang, Lean Fu, and Fei Wang. Deep recurrent neural network with multi-scale bi-directional propagation for video deblurring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3598–3607, 2022. 3

Single Image Defocus Deblurring via Implicit Neural Inverse Kernels (Supplemental Material)

1. Proof of Proposition 1

Consider an image space $\mathbb{R}^{H_0\times W_0}$. For a PSF $\boldsymbol{k} \in \mathbb{R}^{H\times W}$ with $H \ll H_0$ and $W \ll W_0$, its inverse kernel $\boldsymbol{k}^\dagger$ is defined in the frequency domain as follows: for $0 \le \omega_x < H_0, 0 \le \omega_y < W_0$,

$$\mathcal{F}(\boldsymbol{k}^\dagger)[\omega_x,\omega_y] = \begin{cases} \frac{1}{\mathcal{F}(\boldsymbol{k})[\omega_x,\omega_y]}, & \text{if } |\mathcal{F}(\boldsymbol{k})[\omega_x,\omega_y]| \neq 0; \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Consider the standard expansion-based dyadic upsampling operator $\mathcal{U}_s := \uparrow_s$, where the operator $\uparrow_s$: $\mathbb{R}^{H\times W} \to \mathbb{R}^{sH\times sW}$ can be defined in the Fourier domain as follows:

$$(\boldsymbol{k}\uparrow_s)[\omega_x,\omega_y] = \begin{cases} \boldsymbol{k}[\frac{\omega_x}{s},\frac{\omega_y}{s}], & \text{if } \frac{\omega_x}{s},\frac{\omega_y}{s} \in \mathbb{Z}; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In the remaining discussion, any index $[\omega_1,\omega_2]$ outside the range $[0,H_0-1]\times[0,W_0-1]$ is defined as $[\omega_1 \bmod H_0, \omega_2 \bmod W_0]$. Then, we have

$$\mathcal{F}(\boldsymbol{k}\uparrow_s)[\omega_x,\omega_y] = \mathcal{F}(\boldsymbol{k})[s\omega_x, s\omega_y]. \tag{3}$$

By the definition of inverse kernel, we have then

$$\begin{aligned} &\mathcal{F}((\boldsymbol{k}\uparrow_s)^\dagger)[\omega_x,\omega_y] \\ &= \begin{cases} \frac{1}{\mathcal{F}(\boldsymbol{k})[s\omega_x,s\omega_y]}, & \text{if } |\mathcal{F}(\boldsymbol{k})[s\omega_x,s\omega_y]| \neq 0; \\ 0, & \text{otherwise} \end{cases} \\ &= \mathcal{F}(\boldsymbol{k}^\dagger)[s\omega_x,s\omega_y] \\ &= \mathcal{F}((\boldsymbol{k}^\dagger)\uparrow_s)[\omega_x,\omega_y]. \end{aligned} \tag{4}$$

Thus, we have

$$(\boldsymbol{k}\uparrow_s)^\dagger = (\boldsymbol{k}^\dagger)\uparrow_s . \tag{5}$$

That is, the inverse kernel and the upsampling operator are commutative. Suppose that $\boldsymbol{k}^\dagger$ can be expressed as a linear combination over a set of atoms $\boldsymbol{V} = \{\boldsymbol{v}_n\}_{n=1}^N$:

$$\boldsymbol{k}^\dagger = \sum_{n=1}^N w_n \cdot \boldsymbol{v}_n, \tag{6}$$

Then, we have

$$(\boldsymbol{k}\uparrow_s)^\dagger = (\boldsymbol{k}^\dagger)\uparrow_s = (\sum_{n=1}^N w_n\cdot\boldsymbol{v}_n)\uparrow_s = \sum_{n=1}^N w_n\cdot(\boldsymbol{v}_n\uparrow_s). \tag{7}$$

Remark 1. *For interpolation-based upsampling operators, they usually can be expressed as*

$$\mathcal{U}_s(\boldsymbol{k}) = (\boldsymbol{k}\uparrow_s) * \boldsymbol{h}_s, \tag{8}$$

for some low-pass filter $\boldsymbol{h}_s$ whose size is related to s. For instance, $\boldsymbol{h}_s = [\frac{1}{2}, 1, \frac{1}{2}]$ for 1D linear interpolation on $s = 2$. Based on (5) and that $(\boldsymbol{a}*\boldsymbol{b})^\dagger = \boldsymbol{a}^\dagger * \boldsymbol{b}^\dagger$, we have a generalized commutativity between inverse kernel and upsampling:

$$\big(\mathcal{U}_s(\boldsymbol{k})\big)^\dagger = (\boldsymbol{k}\uparrow_s)^\dagger * \boldsymbol{h}_s^\dagger = \big((\boldsymbol{k}^\dagger)\uparrow_s\big) * \boldsymbol{h}_s^\dagger. \tag{9}$$

As a result, in analog to (7), we have a generalized span:

$$(\mathcal{U}_s(\boldsymbol{k}))^\dagger = \sum_{n=1}^N w_n\cdot(\boldsymbol{v}_n\uparrow_s * \boldsymbol{h}_s^\dagger). \tag{10}$$

Such a generalized form is implicitly implemented in our INR-based model.

2. Additional Analysis

2.1. Influence of Atom Size and Atom Number

We investigate the impact of two main hyper-parameters, maximum atom size and atom number, in our INR-based model, by fixing one and varying the other. Concretely, we vary the maximum atom size to 7×7, 9×9, 11×11, 13×13, 15×15, 17×17, 19×19, respectively, with a fixed atom number of 10, and we vary the atom number to 2, 4, 6, 8, 10, 12, 14, 16, respectively, with a fixed maximum atom size of 15×15. The results plotted in Fig. 1 show that (a) when the atom size or atom number is too small, the performance of our INIKNet decreases (noticeable on RealDOF), probably due to under-fitting; (b) when setting the two hyper-parameters to larger values than the ones used in our experiments, the performance of INIKNet even has certain increase in some cases; (c) when the atom number is set too large, INIKNet performs slightly worse, probably due to overfitting; and (d) within a reasonable range of the hyper-parameters, INIKNet performs stably well.

2.2. Deblurring via INR Atoms vs. Non-INR Atoms

See Fig. 2 for some deblurred images on the CUHK dataset using INIKNet and its non-INR version respectively.

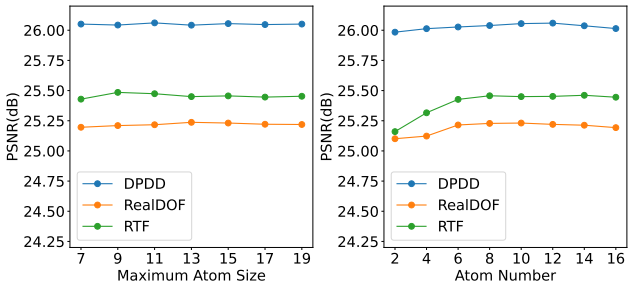


Figure 1: PSNR w.r.t. max atom size and atom number.

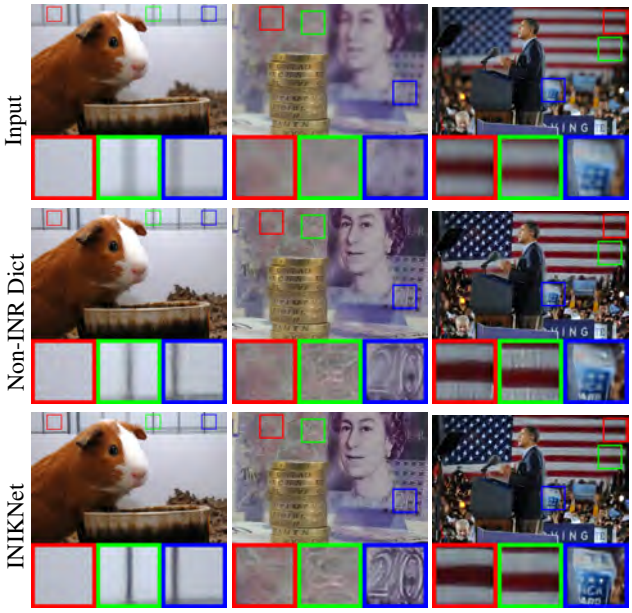


Figure 2: Deblurred image using INR-based model vs. non-INR model on CUHK dataset (without ground-truths). Zoom in for better views.

Leveraging INR-based inverse kernel atoms, INIKNet can recover cleaner textures with fewer artifacts.

2.3. Polar vs. Grid Coordinates in INR

Different from many existing works, our approach utilizes polar coordinates rather than grid coordinates as the input of INR. This does lead to some performance improvement (though not our main contribution), as demonstrated in Table 1.

Input	DPDD			RealDOF			RTF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Grid	26.021	0.801	0.191	25.173	0.758	0.292	25.393	0.825	0.223
Polar	26.055	0.803	0.185	25.231	0.765	0.287	25.450	0.834	0.215

Table 1: Results of grid/polar coordinates used for INR-based atoms in INIKNet.

2.4. Performance Comparison to MIMOUNet

MIMOUNet [1] is a popular milt-scale NN for dynamic scene deblurring which employs asymmetric feature fusion to directly aggregate features of different scales. Table 2 presents the results of MIMOUNet trained on DPDD under the same training setting as ours. The MIMOUNet shows close performance to our INIKNet, but using a much larger model. In addition, its generalization performance on RealDOF and RTF is not as good as that of INIKNet, particularly that the PSNR gap is more than 0.8dB on RTF. These results again demonstrate the effectiveness of our approach.

Method	DPDD			RealDOF			RTF			#Par.
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	(M)
MIMOUNet	25.951	0.798	0.187	24.851	0.746	0.348	24.520	0.795	0.297	16.11
INIKNet	26.055	0.803	0.185	25.231	0.765	0.287	25.450	0.834	0.215	1.98

Table 2: Comparison with MIMOUNet trained on DPDD.

3. Additional Qualitative Results

In addition to those in the main paper, we provide extensive visual comparisons in Fig. 3, 4, 5, 6, 7, 8. When our INIKNet model is trained on DPDD, it not only performs well on the corresponding test split (see Fig. 3), but also shows superior generalization performance on RealDOF (see Fig. 4) and CUHK (see Fig. 5) both of which have a large domain gap against DPDD. Same thing happens when training on LFDOF (see Fig. 6, 7, 8). Under some extreme situations with severe blur, most methods fail while ours still works. Note that although the DRBNet trained on LFDOF achieves comparable performance to our INIKNet on LFDOF, its generalization performance to unseen patterns on CUHK is not as good as that of INIKNet (see Fig. 7, 8). These additional results further demonstrate the superiority of our proposed approach.

References

[1] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4641–4650, 2021. 2

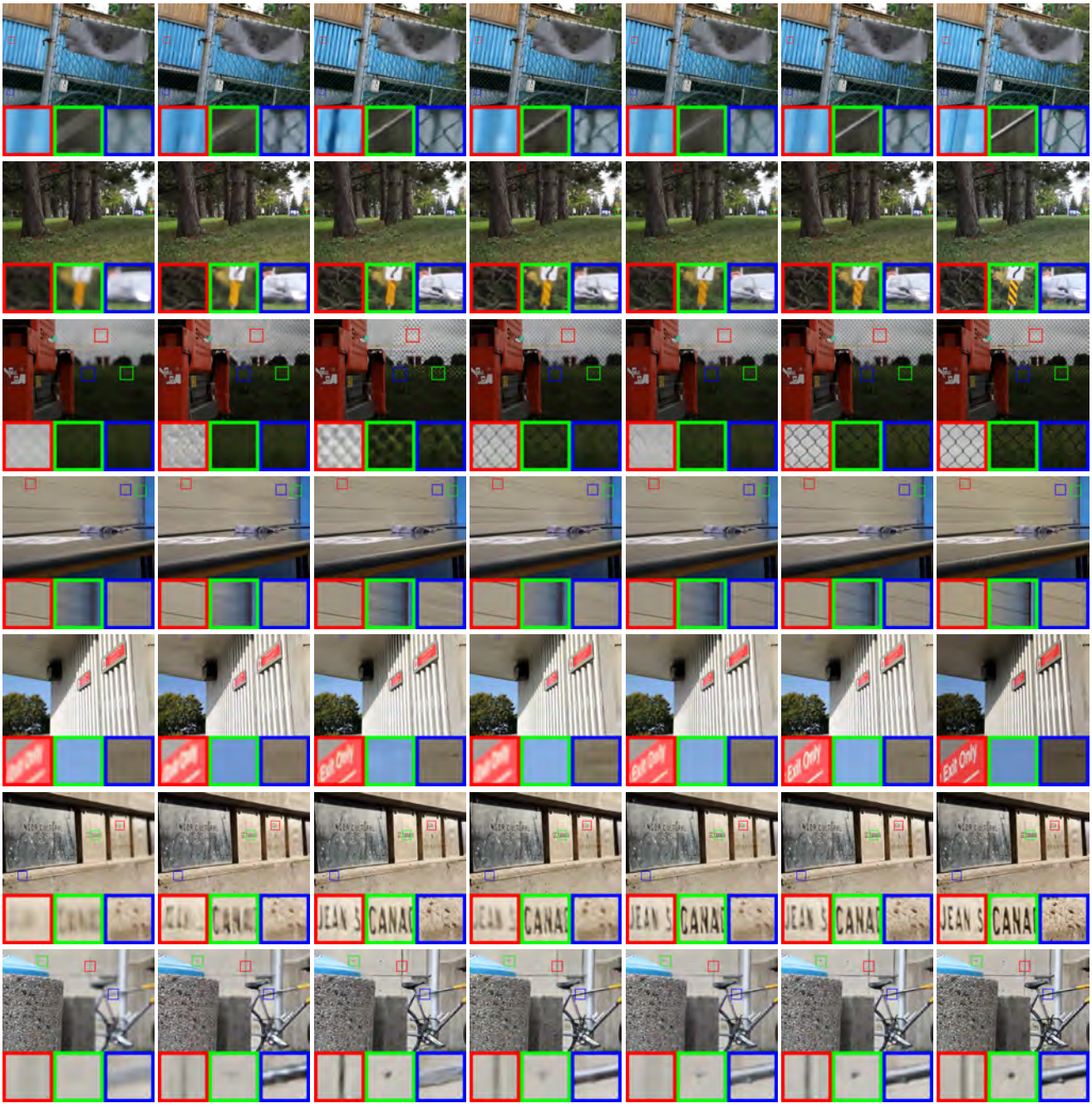
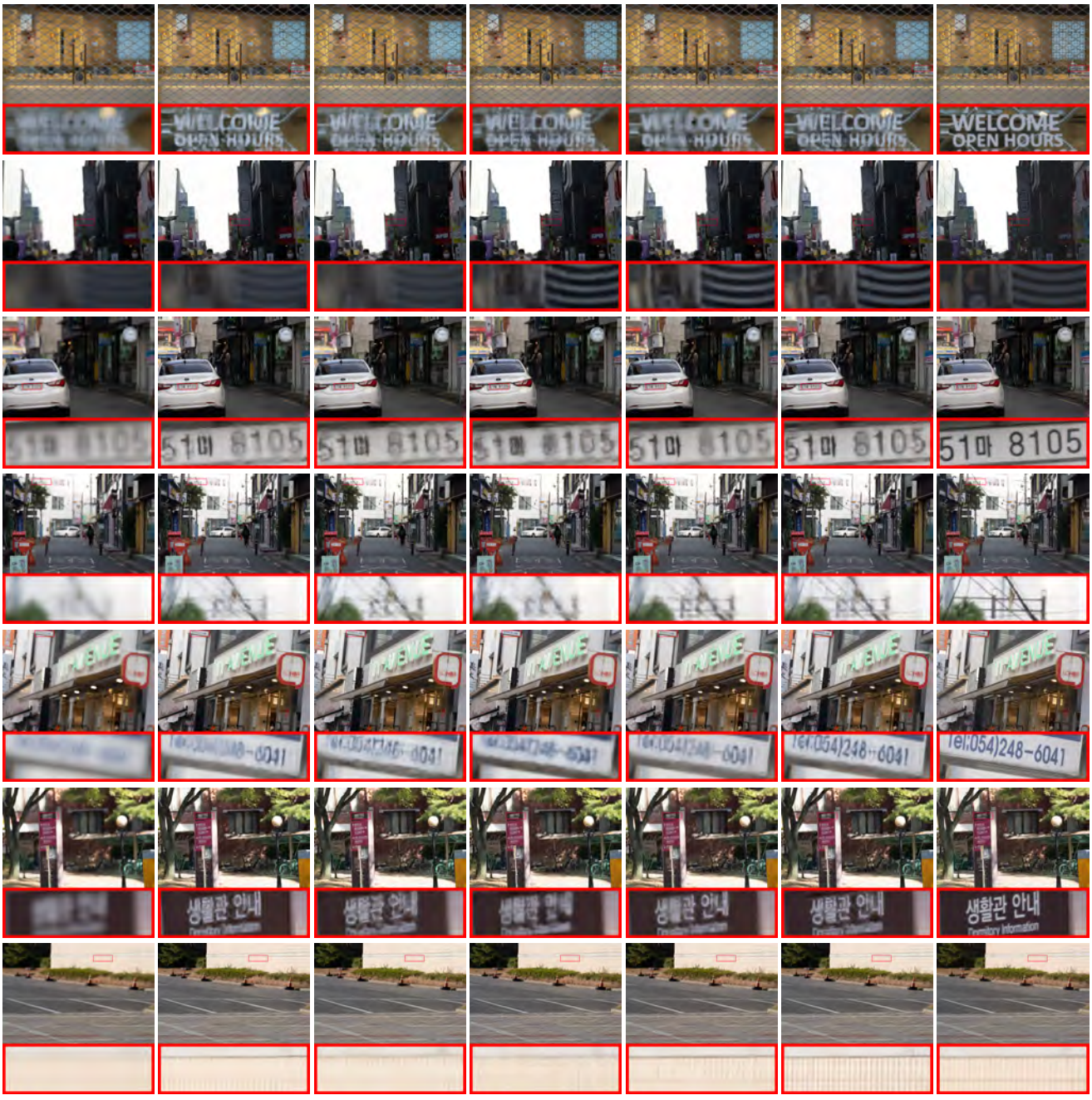


Figure 3: Results on DPDD using models trained on DPDD. Zoom in for better views.



Input IFANet KPAC GKMNet DRBNet INIKNet Ground Truth

Figure 4: Results on RealDOF using models trained on DPDD. Zoom in for better views.

Figure 5: Results on CUHK dataset using models trained on DPDD. Zoom in for better views.

Figure 6: Results on LFDOF using models trained on LFDOF. Zoom in for better views.

Figure 7: Results on CUHK dataset (without ground truths) using models trained on LFDOF. Zoom in for better views.

Figure 8: Results on CUHK dataset (without ground truths) using models trained on LFDOF. Zoom in for better views.