Learning to Predict Decomposed Dynamic Filters for Single Image Motion Deblurring

Zhiqiang Hu[1](✉) and Tao Yu[2]

[1] Yokohama, Japan
zhiqianghu2021@gmail.com
[2] Tokyo Institute of Technology, Tokyo, Japan
yutao@mobile.ee.titech.ac.jp

Abstract. This paper tackles the large motion variation problem in the single image dynamic scene deblurring task. Although fully convolutional multi-scale-based designs have recently advanced the state-of-the-art in single image motion deblurring. However, these approaches usually utilize vanilla convolution filters, which are not adapted to each spatial position. Consequently, it is hard to handle large motion blur variations at the pixel level. In this work, we propose Decomposed Dynamic Filters (DDF), a highly effective plug-and-play adaptive operator, to fulfill the goal of handling large motion blur variations across different spatial locations. In contrast to conventional dynamic convolution-based methods, which only predict either weight or offsets of the filter from the local feature at run time, in our work, both the offsets and weight are adaptively predicted from multi-scale local regions. The proposed operator comprises two components: 1) the offsets estimation module and 2) the pixel-specific filter weight generator. We incorporate the DDF into a lightweight encoder-decoder-based deblurring architecture to verify the performance gain. Extensive experiments conducted on the GoPro, HIDE, Real Blur, SIDD, and DND datasets demonstrate that the proposed method offers significant improvements over the state-of-the-art in accuracy as well as generalization capability. Code is available at: https://github.com/ZHIQIANGHU2021/DecomposedDynamicFilters.

1 Introduction

Dynamic scene motion deblurring aims to rehabilitate an original sharp image from a blurry image caused by camera shakes, moving objects, or low shutter speeds. Blur artifacts significantly degrade the quality of captured images, which is harmful to many high-level vision applications, *e.g.,* face recognition systems, surveillance, and autonomous driving systems. Therefore, the accurate and efficient technique of eliminating blurring artifacts and recovering sharp images is

Z. Hu—Independent Researcher.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-26313-2_24.

L. Wang et al. (Eds.): ACCV 2022, LNCS 13843, pp. 390–408, 2023.
https://doi.org/10.1007/978-3-031-26313-2_24

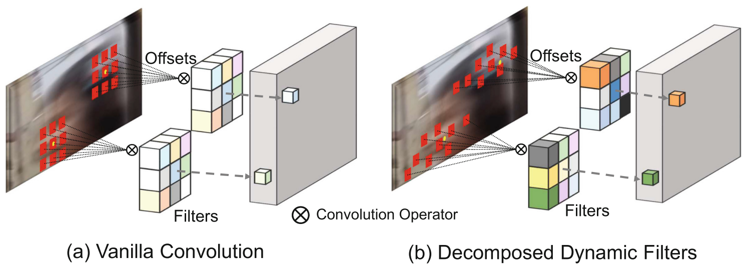


Fig. 1. The vanilla convolution and the proposed DDF. As for DDF, both the offsets and filter weight are adaptively predicted from multi-scale local regions. The red squares are the sampling positions and different colors of cubes indicate the spatially-varying filters of DDF. (Color figure online)

highly desired. To handle the blind motion deblurring problem, many conventional approaches attempt to estimate the blur kernel via some hand-crafted priors [1–7]. However, estimating a satisfactory blur kernel remains a challenging computer vision problem. Such hand-crafted priors can hardly generalize well to complex real-world examples, which results in degraded performance. To address these challenges, many deep learning-based approaches [8–16] try to utilize the neural network to deblur images in an end-to-end manner and significantly improve the performance. In particular, among the architectures, the coarse-to-fine scheme has been widely employed to restore the blurred image at either multiple scales (MSs) [8–10,12] or multiple patch (MPs) levels [11,13–16]. However, these methods usually employ a vanilla convolution filter as the base module, which suffers from two main issues in motion deburring tasks:

1) It is difficult to handle large variations in motion magnitude. More specifically, the identical geometric shape or receptive fields of vanilla convolution filters are applied to different pixels of an image. However, the magnitude of motion blur appears diversely in different regions (*e.g.,* moving vehicle pixels vs. sky). To tackle this problem, extremely deep networks and multi-scale architectures have been exploited to enhance the generalization ability to solve the large variance of motion problems. Consequently, this kind of approach suffers from ultra-heavy computational complexity and is hard to be deployed on lightweight devices, *e.g.,* smartphones and self-driving cars.
2) The weights of the vanilla convolution filters are also content-agnostic as shown in Fig. 1(a), regardless of the texture information of the local region. A spatially shared filter could be sub-optimal for the task of extracting feature across all pixels. In addition, once the network has been trained, the identical filters are utilized across different images, leading to ineffective feature extraction results.

To tackle the problems mentioned above, in this work, we focus on the design of efficient and adaptive filtering modules, dubbed as Decomposed Dynamic Filter (DDF), which is illustrated in Fig. 1(b). DDF decouples a conventional convolution filter into offsets and weights adaptive filters. The proposed DDF consists of two major components:

1) The offsets estimation module, which could learn from local multi-scale features, and generate the optimal filter offsets. The proposed module can be trained end-to-end without explicit supervision. The deformable convolution [17,18] has been proposed to adapt to local image geometry and enlarge the receptive field without sampling extra pixels. However, the offsets estimator of deformable convolution is merely a single-layer structure. Consequently, it does not perform well in large motion situations, which frequently occur in dynamic scenes. To address this issue, we propose a novel adaptive offsets estimation module, that can generate pixel-level representative offsets across the multi-scale spatial space. The proposed offsets estimator naturally solves the large motion problem by capturing long-range dependencies from multi-scale feature regions.
2) The pixel-specific filter weight generator. The weight of DDF is dynamically reconstructed by a linear combination of filter basis with assembling coefficients. Both the filter basis and the assembling coefficients are self-adaptively learned from multi-scale local regions by lightweight sub-networks. In contrast to the dynamic filter-based method [19,20], which explicitly predicts all pixel-specific filter weights from feature maps, our method achieves a better tradeoff between performance and memory usage. Finally, the learned weight and offsets are combined as a dynamic filter to extract the adaptive feature for the motion deblurring task. Deploying the proposed DDF for the motion deblurring network enables us to design a compact structure for the network, which a conventional network [9,11,13] cannot achieve without stacking a large number of subnet blocks. Overall, our contributions can be summarized as follows:
   - We proposed a novel adaptive operator module DDF, which is capable of learning the sampling positions and filter weights at the same time. Adaptively solve the motion variation problem. Moreover, the dynamic feature of DDF enables us to design an effective network structure without sacrificing accuracy.
   - We conducted extensive ablation evaluations on multiple benchmarks and confirmed that the spatial adaptability of DDF could empower various vanilla networks *e.g.,* U-Net [21], and achieve state-of-the-art performance.
   - To consolidate the generalization capability of DDF, and verify its effectiveness as a plug-and-play adaptive operator, we also plug it onto baseline networks for the real image noise removal task, which significantly improves the performance.

2 Related Works

2.1 Single-Scale Deep Image Deblurring

The **single-scale** approaches [22–24] aim to recover blurred images in an end-to-end manner. For instance, DeblurGAN [22], the adversarial learning algorithm, has been adopted with multiple residual blocks to restore the sharp image. DeblurGAN-v2 [23] advanced the DeblurGAN [22] by employing a much deeper architecture with an encoder-decoder architecture. However, the GAN-based methods often introduce unexpected artifacts into the image and make it hard to handle large motion situations. Yuan *et al.* [24] utilized optical flow information to guide deformable convolutions [18] offsets generation process. However, optical flow information is not always available for real-world applications.

2.2 Multi-scale, Multi-patch-Based Approaches

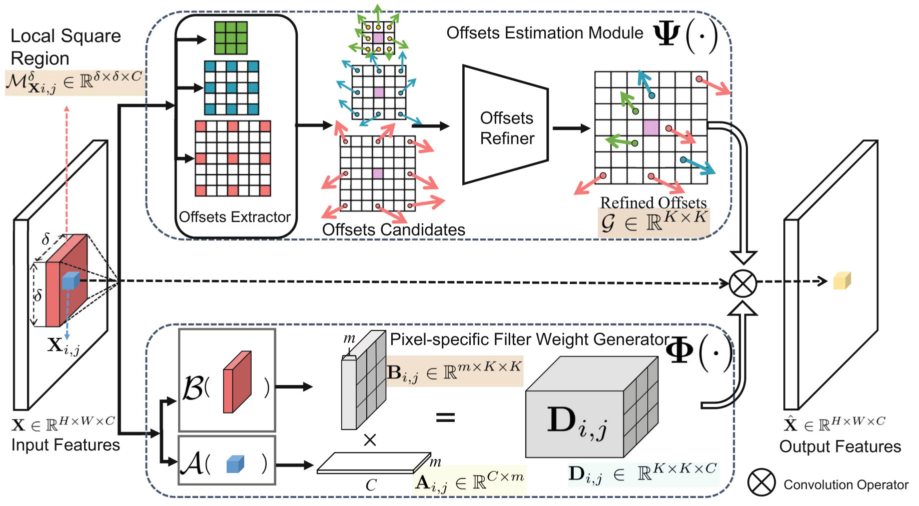
Multi-scale approaches have been verified to be an effective direction in image restoration scenarios. The pioneering work, Nah *et al.* [8] proposed a multi-scale deblurring network, which initiates from a coarse scale and then progressively deblurs the image at a finer scale. Tao *et al.* [9] proposed SRN, a scale-recurrent network responsible for aggregating features from different scales. The motion information from the previous coarser scale can be utilized for the following processing. Cho *et al.* [25] proposed MIMO-UNet, which employs a multi-scale U-Net structure to deblur the image in a coarse-to-fine strategy. The approach in [10] proposed a sharing method that takes a different level of blurs in each stage into consideration. Zhang *et al.* [11] introduced a multi-patch hierarchical scheme (DMPHN) to keep spatial resolution without any image down-sampling operation. The MPRNet [13] combines the **multi-patch** hierarchical structure with a global attention mechanism to further advance the state-of-the-art.

However, these methods usually adopt vanilla convolution kernels and consist of multiple large subnetworks, which lead to a long process time. Recently, **vision transformer (ViT)** with the ability of long-range dependency modeling, has shown promising performance in image restoration tasks [26–28]. However, as suggested in [29], the execution time of SwinIR [28] is approximately 1.99s on GoPro dataset, which is unacceptable for real-time applications. In contrast, we attempt to empower the lightweight network with pixel-adaptive ability, resulting in a lighter network for real-world applications.

2.3 Dynamic Filters

Image level dynamic filters predict filter weight based on input features at the image level. In particular, DyNet [30], DynamicConv [31], and CondConv [32] predict coefficients to combine several expert filters by employing attention mechanisms. However, the dynamic weights are generated at the image scale, making it hard to handle the motion deblur problem, which appears at pixel-level.

Pixel level dynamic filters [17–20,33–42] further extend the adaptiveness to the spatial dimension by using a per-pixel dynamic filter. The filter weights are dynamically predicted based on the input features. Deformable convolutions [17,18] attempt to learn an offsets map at position level, to adapt to the geometric variations, while fixing the weight in kernels. Su *et al.* [19] proposed to adaptively generate pixel-specific [20] filters on an input image. CARAFE [33,34] proposed an adaptive operator for feature map upsampling, where an auxiliary branch is



Local Square Region
$\mathcal{M}^{\delta}_{\mathbf{X}_{i,j}} \in \mathbb{R}^{\delta\times\delta\times C}$
Offsets Extractor
Offsets Candidates
Offsets Estimation Module $\Psi(\cdot)$
Offsets Refiner
Refined Offsets
$\mathcal{G} \in \mathbb{R}^{K\times K}$
δ
δ
$\mathbf{X}_{i,j}$
$\mathbf{X} \in \mathbb{R}^{H\times W\times C}$
Input Features
$\mathcal{B}($ $)$
$\mathcal{A}($ $)$
m
Pixel-specific Filter Weight Generator $\Phi(\cdot)$
$\mathbf{B}_{i,j} \in \mathbb{R}^{m\times K\times K}$
$=$
$\times$
$\mathbf{D}_{i,j}$
m
C
$\mathbf{A}_{i,j} \in \mathbb{R}^{C\times m}$
$\mathbf{D}_{i,j} \in \mathbb{R}^{K\times K\times C}$
$\hat{\mathbf{X}} \in \mathbb{R}^{H\times W\times C}$
Output Features
$\otimes$ Convolution Operator

Fig. 2. The learning framework of DDF, which consists of two components: 1) The offsets estimator and 2) the pixel-specific filter weight generator.

utilized to predict a 2D filter at each pixel. However, these channel-wise shared filters are hard to capture cross-channel information leading to the sub-optimal result. Furthermore, various dynamic filter-based methods have been applied to facilitate computer vision tasks, such as video frame interpolation [35], video denoising [36], super-resolution [37,38], semantic segmentation [39,40], and point cloud segmentation [41,42].

However, the abovementioned methods directly predict all the parameters of dynamic filters, which require a large amount of memory for restoring the gradient in the backpropagation process. In contrast, our method learns to predict the dynamic pixel-adaptive convolution kernels in a memory-efficient manner and also inherits merits of pixel adaptive paradigm, detailed comparison results are in Sect. 3.4.

3 Our Approach

3.1 Overview

The success of Dynamic Convs [17–20,33,34] suggests that adaptively predicting the weight of filters at run-time improves the accuracy. However, it may lack the ability to dynamically adapt to the geometric variations and local image patch textures, at the same time. To this end, we propose DDF, such an operator enjoys benefits from modeling both geometric variations and local textures. The learning framework of DDF is shown in Fig. 2. In this section, we first revisit the general concept of dynamic convolutions, and then we introduce the architecture of the proposed DDF in detail.

3.2 Decomposed Dynamic Filters

We initiate from introducing the vanilla convolutions to make the definition of the proposed DDF. Let $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ denote the input feature map, where H, W, C represent its height, width, and channel numbers, respectively. $\mathbf{X}_{i,j} \in \mathbb{R}^{C}$ is the feature vector inside $\mathbf{X}$ at position i-th row and the j-th column. We also use $\mathcal{M}^{\delta}_{\mathbf{X}i,j} \in \mathbb{R}^{\delta \times \delta \times C}$ represents the size-δ square region centered at i, j inside cube tensor $\mathbf{X}$. Hence, the conventional vanilla convolution can be expressed as

$$\mathbf{X}'_{i,j} = \mathcal{F}\left(\mathcal{M}^{\delta}_{\mathbf{X}_{i,j}}; \mathbf{\Theta}\right), \tag{1}$$

where we employ $\mathcal{F}$ to represent the convolution operation, $\mathbf{X}'$ is the output feature map, δ refers to the kernel size, and the convolution filter parameter $\mathbf{\Theta}$ remains the same across all the pixels over the image. In contrast, the proposed pixel-adaptive operator DDF is conditioned on the corresponding local region of feature map $\mathbf{X}$, which is formulated as

$$\hat{\mathbf{X}}_{i,j} = \mathcal{F}\left(\mathcal{M}^{\delta'}_{\mathbf{X}_{i,j}}; \mathbf{\Theta}_{DDF} \mid \mathcal{M}^{\delta'}_{\mathbf{X}_{i,j}}\right), \tag{2}$$

where $\hat{\mathbf{X}}$ is the output feature map generated by DDF, the parameter $\mathbf{\Theta}_{DDF}$ of DDF is composed of two independent part: 1) the pixel-specific filter weight $\mathbf{D}$, which consists of a list of spatially-varying filters $\{\mathbf{D}_{i,j}\} \in \mathbb{R}^{K^2 \times C}$, where $i \in \{1, 2, \ldots, H\}, j \in \{1, 2, \ldots, W\}$; 2) the adaptive offsets $\mathbf{S}$, $\{\mathbf{S}_{i,j}\} \in \mathbb{R}^{K^2 \times 2}$, where $i \in \{1, 2, \ldots, H\}, j \in \{1, 2, \ldots, W\}$, and K is the kernel size. To handle large motion variations for all the pixels and enable the DDF to see a larger region of the corresponding feature area. We utilize a set of $K \times K$ atrous convolution filters $\{\boldsymbol{W}^r\}_{r=1}^{n}$ with dilated rate r to extract features for the following filter predicting task. δ' refers to the maximum receptive filed of atrous convolution filters set $\{\boldsymbol{W}^r\}_{r=1}^{n}$, and calculated by $\delta' = K + (K-1)(r-1)$. More specifically, the proposed DDF is represented as

$$\hat{\mathbf{X}}_{i,j,k} = \sum_{(u,v) \in \Delta_K} \mathbf{D}_{i,j,u+\lfloor K/2 \rfloor, v+\lfloor K/2 \rfloor, k} \mathbf{X}_{i+u+\Delta x_{i,j}^{u+\lfloor K/2 \rfloor, v+\lfloor K/2 \rfloor}, j+v+\Delta y_{i,j}^{u+\lfloor K/2 \rfloor, v+\lfloor K/2 \rfloor}, k}, \tag{3}$$

where $\boldsymbol{\Delta}_K \in \mathbb{Z}^2$ indicates the set of sampling positions for convolution operation, written as ($\times$ is Cartesian product) $\boldsymbol{\Delta}_K = [-\lfloor K/2 \rfloor, \cdots, \lfloor K/2 \rfloor] \times [-\lfloor K/2 \rfloor, \cdots, \lfloor K/2 \rfloor]$, and $\{\Delta x_{i,j}, \Delta y_{i,j}\} \in \mathbf{S}_{i,j}$ are the learnable offsets at horizontal and vertical directions, respectively. k is the channel index, since we use depth-wise convolution each kernel in $\mathbf{D}$ has C channels, instead of $C_{\text{in}} \times C_{\text{out}}$. Our work aims to design a filtering operator with content-adaptive property. In contrast to vanilla content-agnostic convolution operator, the proposed dynamic filters leverage two meta branch modules to learn the parameter of $\mathbf{D}_{i,j}$ and $\mathbf{S}_{i,j}$ from the local feature region, which is formulated as follows:

$$\mathbf{D}_{i,j} = \mathbf{\Phi}\left(\mathcal{M}^{\delta'}_{\mathbf{X}_{i,j}}; \theta_D\right) \tag{4}$$

$$\mathbf{S}_{i,j} = \mathbf{\Psi}\left(\mathcal{M}^{\delta'}_{\mathbf{X}_{i,j}}; \theta_S\right), \tag{5}$$

where $\mathbf{\Phi}(\cdot)$ and $\mathbf{\Psi}(\cdot)$ are the meta generation networks parameterized by θ, which are responsible for the filter weight learning and offsets learning, respectively. We will give more detailed information in the following sections.

3.3 The Offsets Estimation Module

The deformable convolution [17,18] merely employs a one-layer vanilla convolution with an identical receptive field to estimate the offsets map, for the entire input feature. However, the same receptive field cannot handle large motion variation for all the pixels, leading to sub-optimal offsets estimation results. Furthermore, the meaningful context information is hard to be captured with the limited receptive field. To address this issue, we propose a multi-scale dynamic offsets estimator, which enables the DDF to see a larger region of the corresponding feature area and generate the optimal offsets. Our offsets estimator $\mathbf{\Psi}(\cdot)$ is composed of two parts: the offsets extractor and the offsets refiner arranged in sequence.

The Offsets Extractor
The atrous convolution has been verified to be a powerful operator for enlarging receptive field. To this end, our offset estimator $\mathbf{\Psi}(\cdot)$ first utilize a set of $K \times K$ atrous convolution filters $\{\boldsymbol{W}^r\}_{r=1}^{n}$with dilated rate r to extract n set corresponding multi-scale offset $\hat{\mathbf{S}} = \left\{\hat{\mathbf{S}}_{i,j}^{0}, \ldots, \hat{\mathbf{S}}_{i,j}^{n-1} \mid \hat{\mathbf{S}}_{ij} \in \mathbb{R}^{K\times K\times 2}\right\}$ along with the modulation scalar $\mathbf{\Delta}m = \left\{\mathbf{\Delta}m_{i,j}^{0}, \ldots, \mathbf{\Delta}m_{i,j}^{n-1} \mid \mathbf{\Delta}m_{ij} \in \mathbb{R}^{K\times K}\right\}$. Because the modulation scalar $\mathbf{\Delta}m$, which was introduced in Deformable ConvNets v2, could evaluate the reliability of each $K \times K$ offsets, so we feed them into learnable guided refiner to decide the final offsets $\mathbf{S}_{ij}$ for the position i, j, from the multi-scale candidate offsets.

The Offsets Refiner
To adaptively select the offsets from generated N-set candidates, we also design a sub-module namely offsets refiner. Intuitively, the larger confidence value indicates a better offset estimation result. To this end, the learnable guided refiner decides the final offsets by selecting the one with the maximum confidence value from candidates. We use $\mathcal{G} \in \mathbb{R}^{K\times K}$ to denote the refined index, given the candidate modulation scalar set $\mathbf{\Delta}m \in \mathbb{R}^{K\times K\times n}$. For each position (u, v) in the spatial domain, we have

$$\mathcal{G}_{u,v} = \text{argmax}\left(\mathbf{\Delta}m_{u,v}^{0}, \mathbf{\Delta}m_{u,v}^{1}, \cdots, \mathbf{\Delta}m_{u,v}^{n-1}\right), \tag{6}$$

where argmax$(\cdot)$ generates the index for the maximum value. So values in the refined index range from 0 to $n-1$ and indicate the index of offsets, which should be chosen for the corresponding positions. However, the argmax$(\cdot)$ is not continuous, so the gradient cannot be achieved. To solve this problem, we employ the softmax with temperature to obtain the gradient for argmax$(\cdot)$ through backward propagation. As detailed in Eq. 7, ζ is the extra noise sampled from Gumbel $(0, 1)$ distribution, and τ is the temperature parameter which controls the distribution, and when the τ increases, the distribution becomes more uniform, as the τ approaches 0, the distribution becomes one-hot.

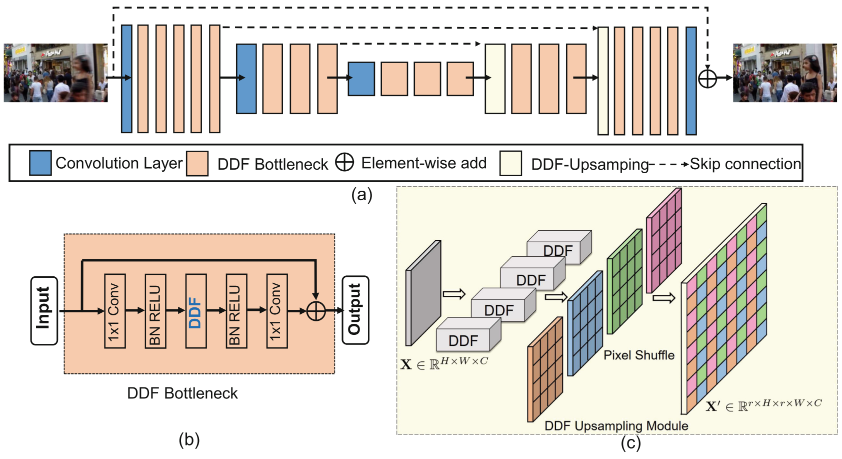


Fig. 3. The architecture of DDF-UNet (a) with DDF Bottleneck module (b), and DDF-Upsampling module (c). The DDF-UNet is a unified framework, suitable for various image restoration tasks, *e.g.,* real world image denoise.

$$\mathcal{G}^j_{u,v} = \frac{e^{\left(\mathcal{G}^j_{u,v}+\zeta\right)/\tau}}{\sum_{m=0}^{n-1} e^{\left(\mathcal{G}^m_{u,v}+\zeta\right)/\tau}} \quad j \in [0, n-1] \tag{7}$$

Therefore, the refiner index module can be trained by the loss function and yield the optimal offsets with respect to different scales for each pixel.

3.4 The Pixel-Specific Filter Weight Generator

The previous dynamic filter-based methods proposed to generate pixel-adaptive filter weight from the local feature. However, generating such a large number of filters ($C_{in} \times C_{out} \times K \times K$) cause extremely large memory usage, where C_{in} and C_{out} are the input channel number and output channel number, respectively. Since each gradient of pixel filter must be saved into memory when doing back-propagation (*e.g.,* input feature map with size $H \times W \times C_{in}$, the gradients size is $H \times W \times C_{in} \times C_{out} \times K \times K$). Thus, the image-level dynamic filters CondConv [32], DYconv [31] can hardly extend to pixel-level filters.

Motivated by the observation that the traditional convolution filter can be well represented by a linear combination of low-rank filter basis along with decomposition coefficients without losing performance [43], we propose the pixel-specific filter weight generator $\mathbf{\Phi}(\cdot)$. The generated dynamic filter $\mathbf{D}_{i,j} \in \mathbb{R}^{K\times K\times C}$, which could decouple into pixel-adaptive basis $\mathbf{B}_{i,j} \in \mathbb{R}^{m\times K\times K}$ and dynamic coefficients $\mathbf{A}_{i,j} \in \mathbb{R}^{C\times m}$, formulated as $\mathbf{D}_{i,j} = \mathbf{B}_{i,j}\mathbf{A}_{i,j}$, m is a pre-defined small value, *e.g.,* $m = 4$.

The work in [43] leaves the coefficients $\mathbf{A}$ as the global parameter shared throughout the image, however, the globally shared filters are hard to capture cross-channel information. Hence, we propose to enable coefficients $\mathbf{A}$ to be a pixel-specific dynamic parameter and utilized two layers of MLP (MultiLayer Perceptron) to capture the cross-channel variations. The pixel-adaptive basis $\mathbf{B}$ and dynamic coefficients $\mathbf{A}$ are generated as follows:

$$\mathbf{B}_{i,j} \in \mathbb{R}^{m\times K\times K} = \mathcal{B}\left(\mathcal{M}^{\delta}_{\mathbf{X}_{i,j}}; \theta_B\right); \mathbf{A}_{i,j} \in \mathbb{R}^{C\times m} = \mathcal{A}\left(\mathbf{X}_{i,j}; \theta_A\right), \tag{8}$$

where, $\mathcal{B}(\cdot)$ and $\mathcal{A}(\cdot)$ are the generation network with parameters θ_B and θ_A, respectively. In our experiment, $\mathcal{B}(\cdot)$ is implemented by one 1×1 convolution followed by a single layer $K \times K$ atrous convolution with the dilated rate $r = n$, the same as offsets estimator, for the purpose of capturing long-range dependencies. As for $\mathcal{A}(\cdot)$, we only utilized two layers of MLP to generate the coefficients $\mathbf{A}$. With the help of the decomposed coefficients, the number of parameters is significantly reduced from $(HWKKC)$ to $(CKKm + mHW)$. Specifically, m can be set to a small value, *e.g.,* $m = 1$, relatively yielding a $(HWKKC)/(HWKK + HW) \approx (C)$ times reduction of parameters.

4 Experiments

4.1 Datasets

To evaluate the proposed method, we conduct extensive experiments on three image deblurring benchmark datasets: 1) GoPro [8] which consisting of 3, 214 image pairs, in which 2,103 pairs are utilized for training our model and the rest 1111 pairs are for testing, 2) HIDE [44] contains 2,025 pairs of images, all of which are used for testing, and 3) RealBlur [45] dataset which consisting of 980 image pairs in RealBlur-R and RealBlur-J test sets, respectively. We train our models on 2,103 pairs of blurry and sharp images from the GoPro dataset, then test against all the test sets from three datasets. The average values of PSNR and SSIM [46] are utilized for performance comparison.

4.2 Implementation Details

Setting. We train our model for 2000 epochs by employing Adam optimizer [47] with default setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is set to be 2×10^{-4} and exponentially decays to 0 using power 0.3. All parameters are initialized using Xavier normalization. The batch size is set to 16. We randomly crop the images into 256×256 patches for training and also utilize horizontal flip and rotation for image augmentation. Experiments are conducted on the server with an intel E5-2690 CPU, and 4X NVIDIA Tesla v100 GPUs.

Network Architecture. The architecture of U-Net-based network [21] is illustrated in Fig. 3(a). We also propose an extension operator DDF-Bottleneck, which is shown in Fig. 3(b), and a DDF-Upsampling module, as shown in Fig. 3(c). According to the upsampling rate r the number of r^2 DDF is used. We utilized pixel-shuffle [48] to assemble the output features as an upsampling layer.

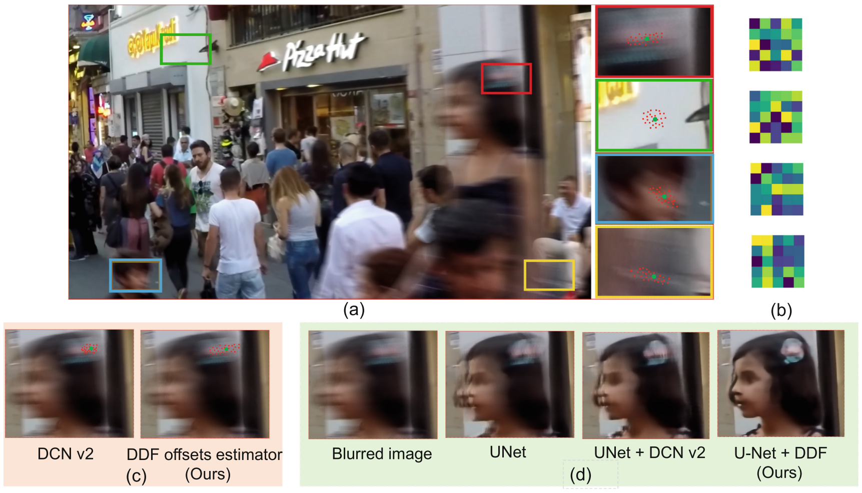


Fig. 4. The illustration of the learned offset (a) and filters (b). As shown in (a), for the slightly blurred regions, the estimated offsets are approximately uniformly distributed over the local area, while for the large motion regions, the sampling geometry of the filter is adjusted adaptively according to the motion blur patterns. The weights of the filter are also adaptively changed to each local region (b), which consolidates that our DDF has indeed learned spatial adaptability. (c) Visualization of the learned offsets of DCN v2 [18] and proposed offset estimator, compared to DCN v2, our offset estimator could capture a larger region of context information along the motion pattern. (d) Comparison results of original UNet, UNet equipped with DCN v2, and proposed DDF-UNet, respectively. Thanks to the spatial-adaptive capability of DDF, the large motion-blurred patch is better recovered than original-UNet and UNet-DCN v2.

4.3 Evaluation of DDF

Because our DDF module is a plug-and-play operator, to verify its effectiveness in improving motion deblurring accuracy, we plug it into various deblurring architectures, such as U-Net and MIMO-UNet [25]. For U-Net, we evaluate two variants of our model: 1) DDF-UNet which consists of 8 encoder blocks and 8 decoder blocks, respectively, and 2) DDF-UNet+ which consists of 20 blocks for each encoder and decoder, respectively. The ℓ_1 loss is used in our implementation, which is formulated as $L = \frac{1}{N}\sum_{n=1}^{N}\left\|\hat{\mathbf{Y}}^{(n)} - \mathbf{Y}^{(n)}\right\|_1$, where $\mathbf{Y}^{(n)}$ is the n-th corresponding sharp image, $\hat{\mathbf{Y}}^{(n)}$ is the network output, and N is the number of sample images in a mini-batch.

The detailed architecture is shown in Fig. 3(a). For MIMO-UNet and MIMO-UNet+ [25], we replace all the ResBlock in MIMO-UNet [25] with our DDF-Bottleneck module, and also replace the upsampling layer with our DDF-Upsampling module, dubbed as DDF-MIMO and DDF-MIMO+, correspondingly. We also use the same loss function: multi-scale frequency reconstruction (MSFR) loss, from the original work [25]. Embedding the DDF module con-

Table 1. Quantitative comparisons on GoPro [8], HIDE [44], and RealBlur [45] dataset. Our methods and the best results are **highlighted**.

Method	GoPro [8]		HIDE [44]		RealBlur-R [45]		RealBlur-J [45]		Param (M)	MACs (G)	Time (s)
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM			
DeblurGAN [22]	28.70	0.858	24.51	0.871	33.79	0.903	27.97	0.834	N/A	N/A	N/A
DeepDeblur [8]	29.08	0.914	25.73	0.874	32.51	0.841	27.87	0.827	11.72	4729	1.290
Zhang *et al.* [14]	29.19	0.931	–		35.48	0.947	27.80	0.847	N/A	N/A	N/A
DeblurGAN-v2 [23]	29.55	0.934	26.61	0.875	35.26	0.944	28.70	0.866	N/A	N/A	N/A
SRN [9]	30.26	0.934	28.36	0.915	35.66	0.947	28.56	0.867	8.06	20134	0.736
Shen *et al.* [44]	–		28.89	0.930	–		–		N/A	N/A	N/A
Gao *et al.* [10]	30.90	0.935	29.11	0.913	–		–		2.84	3255	0.316
DBGAN [49]	31.10	0.942	28.94	0.915	33.78	0.909	24.93	0.745	N/A	N/A	N/A
MT-RNN [12]	31.15	0.945	29.15	0.918	35.79	0.951	28.44	0.862	2.6	2315	0.323
DMPHN [11]	31.20	0.940	29.09	0.924	35.70	0.948	28.42	0.860	7.23	1100	0.137
SAPHN [16]	31.85	0.948	29.98	0.930	–		–		N/A	N/A	N/A
SPAIR [50]	32.06	0.953	30.29	0.931	–		28.81	0.875	N/A	N/A	N/A
HINet [15]	32.71	0.959	–		–		–		88.67	2401	0.247
MPRNet [13]	32.66	0.959	30.96	0.939	35.99	**0.952**	28.70	0.873	20.1	10927	1.023
Original-UNet	28.94	0.912	27.42	0.873	32.51	0.898	26.80	0.801	4.1	120	0.031
DDF-UNet (ours)	30.31	0.934	28.34	0.915	33.72	0.914	27.40	0.829	6.5	170	0.041
DDF-UNet+ (ours)	31.02	0.940	28.92	0.917	34.59	0.938	27.62	0.833	13.2	420	0.101
MIMO-UNet [25]	31.73	0.951	29.28	0.921	35.47	0.946	27.76	0.836	6.8	944	0.133
DDF-MIMO (ours)	32.68	0.958	29.72	0.928	35.62	0.948	27.98	0.849	9.2	1100	0.137
MIMO-UNet+ [25]	32.45	0.957	29.99	0.930	35.54	0.947	27.63	0.837	16.1	2171	0.290
DDF-MIMO+ (ours)	**32.89**	**0.961**	**30.99**	**0.931**	**36.10**	**0.952**	**28.76**	**0.874**	19.2	2370	0.320

tributes to remarkable performance gains for both architectures, as detailed in Table 1, *e.g., w/*DDF in DDF-UNet lead to a 1.37 dB improvement in PSNR on the GoPro dataset, a toy example is shown in Fig. 4. For DDF-MIMO, a 0.95 dB gain in PSNR is verified by plugging the DDF onto MIMO-UNet [25] architecture. These results consolidate that, the spatial adaptability of DDF could empower the vanilla network, and achieve better performance.

4.4 Comparisons with State-of-the-Art Methods

We extensively compare our proposed method with state-of-the-art dynamic scene motion deblurring approaches, including DeepBlur [8], DeblurGAN v1, v2, [22,23], SRN [9], Gao *et al.* [10] and DMPHN [11], MPRNet [9] and so on. The quantitative results on the GoPro [8], HIDE [44], and RealBlur [45] test sets are listed in Table 1. The visual comparison results are illustrated in Fig. 5 and Fig. 6. As illustrated in the images, our method outperforms the other approaches in terms of deblurring quality, and handles large dynamic motion blur scenes quite well. It can be seen from the quantitative results that our DDF-MIMO+ outperforms the previous state-of-the-art MPRNet [13], in terms of PSNR, ours DDF-MIMO+ is ranked first, surpassing the best competitor [13]. In terms of inference-time, the proposed DDF-UNet can deblur one image at 41ms. In contrast, the stacked architectures, *e.g.,* SRN [9], DMPHN [11], and SAPHN [16] suffer from expensive computational costs because they need to stack more layers for achieving larger receptive fields. Our method is 17.9× and

Table 2. Ablation study based on GoPro [8] testing set for using different component combinations in DDF-UNet, where DCN v2 represents Deformable Convolution V2 [18]. DDF Offsets means the DDF offsets estimation module, while DDF Weight indicates the DDF pixel-specific filter weight generator. The best result is **highlighted**.

Version	Ori-UNet	DCN v2	DDF Offsets	DDF Weight	DDF-Upsampling	PSNR (dB)
Version 1	✓					28.94
Version 2	✓	✓				29.21
Version 3	✓		✓			29.67
Version 4	✓		✓	✓		30.10
Version 5	✓		✓	✓	✓	**30.31**

Table 3. Real Image Denoising results on the SIDD [51] and DND [52] datasets. Our methods and the best results are **highlighted**.

Method	SIDD [51]		DND [52]	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
DnCNN [53]	23.66	0.583	32.43	0.790
CBM3D [54]	25.65	0.685	34.51	0.851
CBDNet [55]	30.78	0.801	34.51	0.851
RIDNet [56]	38.71	0.914	38.06	0.942
DANet [57]	39.47	0.918	39.59	0.955
SADNet [58]	39.46	0.957	39.59	0.952
CycleISP [59]	39.52	0.957	39.56	**0.956**
MIRNet [60]	39.72	0.959	39.88	**0.956**
MPRNet [13]	39.71	0.958	39.80	0.954
NBNet [61]	39.75	**0.959**	39.89	0.955
Original UNet	37.02	0.895	37.80	0.947
DDF-UNet (ours)	39.62	0.957	39.81	0.954
DDF-UNet+ (ours)	**39.82**	**0.959**	**39.94**	**0.956**

31× faster than SRN [9] and DeepBlur [8], respectively. The experimental results demonstrate the optimal trade-off between the performance and computational complexity of the proposed method.

4.5 Ablation Studies

Analyses of the Spatial Adaptability. As illustrated in the Fig. 4, DDF handles large dynamic motion blur scenes quite well. We observe that the result of UNet suffers from artifacts and deficient deblurring results depicted in Fig. 4(d).

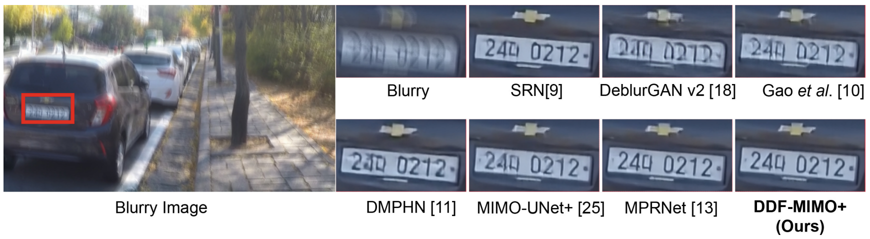


Fig. 5. Visual comparisons for the GoPro dataset [8].

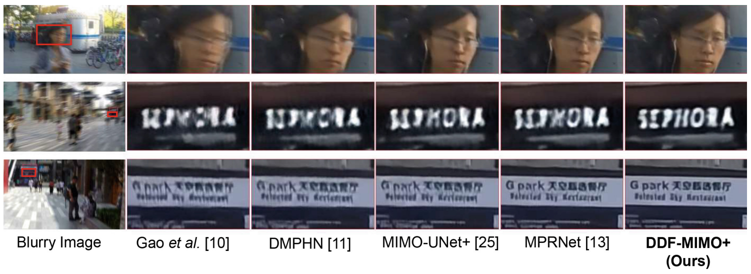


Fig. 6. Visual comparisons for the HIDE [44] dataset.

In contrast, owing to the proposed DDF module, our model is capable of restoring sharper boundaries and richer details in the region containing large motion blur. In addition, according to the visualization of the offsets estimation result in Fig. 4(a), for the background region, the estimated offsets are approximately uniformly distributed in the area, while for the large motion regions, we can see that the sampling geometry of the filter is adjusted adaptively according to the motion blur patterns. Moreover, as shown in Fig. 4(b), the weights of filters are also adaptively changed for each local region, which demonstrates that our DDF has been empowered with spatial adaptability.

The Effectiveness of Offsets Estimator and Refiner. To demonstrate the effectiveness of each component in the proposed DDF including the adaptive offsets prediction module, and the kernel weight perdition module, we compare the results with several versions. The experimental results are listed in Table 2. **Version 1** is the original U-Net, as the baseline. **Version 2** is U-Net equipped with Deformable ConvNets V2 [18], **Version 3** is replaces deformable convolution with the adaptive offsets prediction module. It achieves an increase of 0.46 dB in PSNR, which means that the offsets adaptive adjustment ability is enhanced by the proposed adaptive offsets prediction module. Compared with deformable convolution, which could only capture a limited area of motion pattern as shown in Fig. 4(c), the proposed DDF demonstrates stronger fitting capabilities.

The Effectiveness of Pixel-Specific Filter Weight Generator. To further evaluate the effectiveness of the pixel-specific filter weight generator, we also

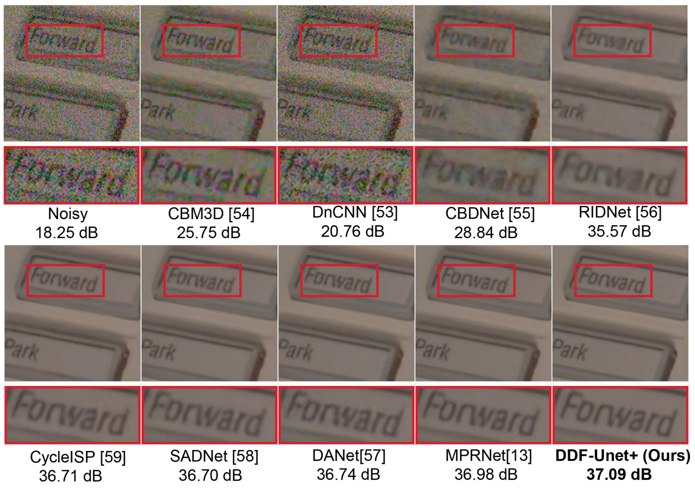


Fig. 7. Visual comparisons for the SIDD [51] dataset.

propose **Version 4** test, which incrementally adds the filter weight generator to DDF. Thanks to the adaptive adjustment of weights, the PSNR has further increased by 0.43 dB.

The Effectiveness of the DDF-Upsampling Module. Finally, we further add the DDF-Upsampling module into the UNet as the **Version 5** test and get a 0.21 dB performance gain, which consolidates the effectiveness of the DDF-Upsampling module.

4.6 Generalization to Real Image Noise Removal Task

To demonstrate the generalization capability of DDF, we also applied the DDF-UNet+ to real-world image noise removal task. The real-world noise removal aims at restoring high-quality images from noisy inputs, which are spatially non-uniformly distributed. Thus, the spatially-adaptive operator is a natural method to solve such kinds of problems. The real-world noise removal dataset SIDD [51] and DND [52] are used for evaluation. We train our model by using 320 high-resolution images of the SIDD dataset and evaluate the test sets of SIDD [51] and DND [52]. The quantitative results are listed in Table 3. We compare DDF-UNet+ with state-of-the-art denoising methods, including the CBM3D [54], CBDNet [55], RIDNet [56], SADNet [58], DnCNN [53], CycleISP [59], NBNet [61], DANet [57], MIRNet [60], and MPRNet [13]. Our DDF-UNet+ achieves a 39.82 dB on PSNR for SIDD [51] dataset, and outperforms the best methods, in our experiments. The visualization results of the SIDD dataset are shown in Fig. 7. The results indicate that the DDF could handle the spatial-variance problem, and improve the performance.

5 Conclusion

In this paper, we proposed a new adaptive plug-and-play operator DDF for the challenging task of handling large motion blur variations across spatial locations. The weights and the offsets of DDF are adaptively generated from the local features by proposed meta networks, which are trained end-to-end without explicit supervision. We have also proposed a U-Net-based architecture powered by our proposed DDF and DDF-Upsampling module. Extensive experimental results demonstrated that the proposed DDF could empower the baseline to achieve better performance. Furthermore, the proposed DDF can be generalized to other computer vision tasks as a plug-and-play adaptive operator, *e.g.,* real image noise removal task, and also achieve state-of-the-art performance.

References

1. Gupta, A., Joshi, N., Lawrence Zitnick, C., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 171–184. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15549-9_13
2. Liu, G., Chang, S., Ma, Y.: Blind image deblurring using spectral properties of convolution operators. IEEE Trans. Image Process. **23**, 5047–5056 (2014)
3. Tran, P., Tran, A.T., Phung, Q., Hoai, M.: Explore image deblurring via encoded blur kernel space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11956–11965 (2021)
4. Ren, D., Zhang, K., Wang, Q., Hu, Q., Zuo, W.: Neural blind deconvolution using deep priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3341–3350 (2020)
5. Chakrabarti, A.: A neural approach to blind motion deblurring. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 221–235. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_14
6. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 769–777 (2015)
7. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. IEEE Trans. Pattern Anal. Mach. Intell. **38**, 1439–1451 (2015)
8. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3883–3891 (2017)
9. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8174–8182 (2018)
10. Gao, H., Tao, X., Shen, X., Jia, J.: Dynamic scene deblurring with parameter selective sharing and nested skip connections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3848–3856 (2019)
11. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5978–5986 (2019)

12. Park, D., Kang, D.U., Kim, J., Chun, S.Y.: Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12351, pp. 327–343. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58539-6_20
13. Zamir, S.W., et al.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14821–14831 (2021)
14. Zhang, J., et al.: Dynamic scene deblurring using spatially variant recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2521–2529 (2018)
15. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: HINet: half instance normalization network for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 182–192 (2021)
16. Suin, M., Purohit, K., Rajagopalan, A.: Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3606–3615 (2020)
17. Dai, J., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)
18. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable ConvNets v2: more deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9308–9316 (2019)
19. Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., Kautz, J.: Pixel-adaptive convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11166–11175 (2019)
20. Zamora Esquivel, J., Cruz Vargas, A., Lopez Meyer, P., Tickoo, O.: Adaptive convolutional kernels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops(2019)
21. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
22. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: DeblurGAN: blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8183–8192 (2018)
23. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: DeblurGAN-v2: deblurring (orders-of-magnitude) faster and better. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8878–8887 (2019)
24. Yuan, Y., Su, W., Ma, D.: Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3555–3564 (2020)
25. Cho, S.J., Ji, S.W., Hong, J.P., Jung, S.W., Ko, S.J.: Rethinking coarse-to-fine approach in single image deblurring. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4641–4650 (2021)
26. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: a general U-shaped transformer for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17683–17693 (2022)

27. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5728–5739 (2022)
28. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: SwinIR: image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1833–1844 (2021)
29. Mao, X., Liu, Y., Shen, W., Li, Q., Wang, Y.: Deep residual Fourier transformation for single image deblurring. arXiv preprint arXiv:2111.11745 (2021)
30. Zhang, Y., Zhang, J., Wang, Q., Zhong, Z.: DyNet: dynamic convolution for accelerating convolutional neural networks. arXiv preprint arXiv:2004.10694 (2020)
31. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11030–11039 (2020)
32. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: CondConv: conditionally parameterized convolutions for efficient inference. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
33. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D.: CARAFE: content-aware reassembly of features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3007–3016 (2019)
34. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D.: CARAFE++: unified content-aware reassembly of features. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 4674–4687 (2021)
35. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive convolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 670–679 (2017)
36. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2502–2510 (2018)
37. Xu, Y.S., Tseng, S.Y.R., Tseng, Y., Kuo, H.K., Tsai, Y.M.: Unified dynamic convolutional network for super-resolution with variational degradations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12496–12505 (2020)
38. Magid, S.A., et al.: Dynamic high-pass filtering and multi-spectral attention for image super-resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4288–4297 (2021)
39. Liu, J., He, J., Qiao, Yu., Ren, J.S., Li, H.: Learning to predict context-adaptive convolution for semantic segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 769–786. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_46
40. He, J., Deng, Z., Qiao, Y.: Dynamic multi-scale filters for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3562–3572 (2019)
41. Xu, C., et al.: SqueezeSegV3: spatially-adaptive convolution for efficient point-cloud segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12373, pp. 1–19. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_1
42. Xu, M., Ding, R., Zhao, H., Qi, X.: PAConv: position adaptive convolution with dynamic kernel assembling on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3173–3182 (2021)

43. Wang, Z., Miao, Z., Hu, J., Qiu, Q.: Adaptive convolutions with per-pixel dynamic filter atom. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12302–12311 (2021)
44. Shen, Z., et al.: Human-aware motion deblurring. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
45. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 184–201. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_12
46. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004)
47. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
48. Shi, W., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883 (2016)
49. Zhang, K., et al.: Deblurring by realistic blurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2737–2746 (2020)
50. Purohit, K., Suin, M., Rajagopalan, A., Boddeti, V.N.: Spatially-adaptive image restoration using distortion-guided networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2309–2319 (2021)
51. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1692–1700 (2018)
52. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1586–1595 (2017)
53. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**, 3142–3155 (2017)
54. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**, 2080–2095 (2007)
55. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1712–1722 (2019)
56. Zhuo, S., Jin, Z., Zou, W., Li, X.: RIDNet: recursive information distillation network for color image denoising. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
57. Yue, Z., Zhao, Q., Zhang, L., Meng, D.: Dual adversarial network: toward real-world noise removal and noise generation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12355, pp. 41–58. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2_3
58. Chang, M., Li, Q., Feng, H., Xu, Z.: Spatial-adaptive network for single image denoising. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 171–187. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58577-8_11

59. Zamir, S.W., et al.: CycleISP: real image restoration via improved data synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2696–2705 (2020)
60. Zamir, S.W., et al.: Learning enriched features for real image restoration and enhancement. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 492–511. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_30
61. Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., Liu, S.: NBNet: noise basis learning for image denoising with subspace projection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4896–4906 (2021)