# Iterative Filter Adaptive Network for Single Image Defocus Deblurring

Junyong Lee      Hyeongseok Son      Jaesung Rim      Sunghyun Cho      Seungyong Lee

POSTECH

{junyonglee, sonhs, jsrim123, s.cho, leesy}@postech.ac.kr

https://github.com/codeslake/IFAN

## Abstract

*We propose a novel end-to-end learning-based approach for single image defocus deblurring. The proposed approach is equipped with a novel Iterative Filter Adaptive Network (IFAN) that is specifically designed to handle spatially-varying and large defocus blur. For adaptively handling spatially-varying blur, IFAN predicts pixel-wise deblurring filters, which are applied to defocused features of an input image to generate deblurred features. For effectively managing large blur, IFAN models deblurring filters as stacks of small-sized separable filters. Predicted separable deblurring filters are applied to defocused features using a novel Iterative Adaptive Convolution (IAC) layer. We also propose a training scheme based on defocus disparity estimation and reblurring, which significantly boosts the deblurring quality. We demonstrate that our method achieves state-of-the-art performance both quantitatively and qualitatively on real-world images.*

## 1. Introduction

Defocus deblurring aims to restore an all-in-focus image from a defocused image, and is highly demanded by daily photographers to remove unwanted blur. Moreover, restored all-in-focus images can greatly facilitate high-level vision tasks such as semantic segmentation [21, 30] and object detection [6, 3]. Despite the usefulness, defocus deblurring remains a challenging problem as defocus blur is spatially varying in size, and its shape also varies across the image.

A conventional strategy [28, 7, 22, 5, 12, 14] is to model defocus blur as a combination of different convolution results obtained by applying predefined kernels to a sharp image. It estimates per-pixel blur kernels based on the blur model and then performs non-blind deconvolution [8, 15, 13]. However, this approach often fails due to the restrictive blur model, which disregards the nonlinearity of a real-world blur and constrains defocus blur in specific shapes such as disc [7] or Gaussian kernels [28, 22, 12, 14].

Recently, Abuolaim and Brown [1] proposed the first end-to-end learning-based method, DPDNet, which does not rely on a specific blur model, but directly restores a sharp image. Thanks to the end-to-end learning, DPDNet outperforms previous approaches for real-world defocused images. They show that dual-pixel data obtainable from some modern cameras can significantly boost the deblurring performance, and present a dual-pixel defocus deblurring (DPDD) dataset. However, ringing artifacts and remaining blur can often be found in their results, mainly due to its naïve UNet [26] architecture, which is not flexible enough to deal with spatially-varying and large blur [37].

In this paper, we propose an end-to-end network embedded with our novel *Iterative Filter Adaptive Network (IFAN)* for single image defocus deblurring. IFAN is specifically designed for the effective handling of spatially-varying and large defocus blur. To handle the spatially-varying nature of defocus blur, IFAN adopts an adaptive filter prediction scheme motivated by recent filter adaptive networks (FANs) [35, 37]. Specifically, IFAN does not directly predict pixel values, but generates spatially-adaptive per-pixel deblurring filters, which are then applied to features from an input defocus blurred image to generate deblurred features.

To efficiently handle large defocus blur that requires large receptive fields, IFAN predicts stacks of small-sized separable filters instead of conventional filters unlike previous FANs. To apply predicted separable filters to features, we also propose a novel *Iterative Adaptive Convolution (IAC)* layer that iteratively applies separable filters to features. As a result, IFAN significantly improves the deblurring quality at a low computational cost in the presence of spatially-varying and large defocus blur.

To further improve the single image deblurring quality, we train our network with novel defocus-specific tasks: defocus disparity estimation and reblurring. The learning of defocus disparity estimation exploits dual-pixel data, which provides stereo images with a tiny baseline, whose disparities are proportional to defocus blur magnitudes [9, 24, 1]. Leveraging dual-pixel stereo images, we train IFAN to pre-

dict the disparity map from a single image so that it can also learn to predict blur magnitudes more accurately.

On the other hand, the learning of reblurring task, which is motivated by the reblur-to-deblur scheme in [4], utilizes deblurring filters predicted by IFAN for reblurring all-in-focus images. For accurate reblurring, IFAN needs to predict deblurring filters that contain accurate information about the shapes and sizes of defocus blur. During training, we introduce an additional network that inverts predicted deblurring filters to reblurring filters and reblurs the ground-truth all-in-focus image. We then train IFAN to minimize the difference between the defocused input image and the corresponding reblurred image. We experimentally show that both tasks significantly boost the deblurring quality.

To verify the effectiveness of our method on diverse real-world images from different cameras, we extensively evaluate the method on several real-world datasets such as the DPDD dataset [1], Pixel dual-pixel test set [1], and CUHK blur detection dataset [27]. In addition, for quantitative evaluation, we present the Real Depth of Field (RealDOF) test set that provides real-world defocused images and their ground-truth all-in-focus images.

To summarize, our contributions include:

- *Iterative Filter Adaptive Network (IFAN)* that effectively handles spatially-varying and large defocus blur,

- a novel training scheme that utilizes the learning of defocus disparity estimation and reblurring, and

- state-of-the-art performance of defocus deblurring in terms of deblurring accuracy and computational cost.

## 2. Related Work

**Defocus deblurring** Most defocus deblurring approaches, including classical and recent deep-learning-based ones, assume a specific blur model and try to estimate a defocus map based on the model. They estimate the defocus map of an image leveraging various cues, such as hand-crafted features [28, 13, 5, 12], learned regression features [7], and deep features [22, 14]. They then utilize non-blind deconvolution methods such as [8, 15, 13] to restore a sharp image. However, their performance is limited due to their restrictive blur models. Recently, Abuolaim and Brown [1] proposed an end-to-end learning-based approach that outperforms previous blur model-based ones. However, their results often have ringing artifacts and remaining blur as mentioned in Sec. 1.

**Filter adaptive networks** FANs have been proposed to facilitate the spatially-adaptive handling of features in various tasks [2, 19, 20, 18, 11, 31, 37, 35, 29, 33]. FANs commonly consist of two components: prediction of spatially-adaptive filters and transformation of features using the predicted filters, where the latter component is called filter
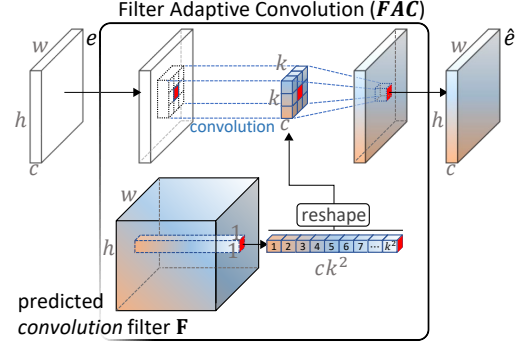


Figure 1. Filter Adaptive Convolution (FAC) [37]

adaptive convolution (FAC). Various FANs have been proposed and applied to different tasks, such as frame interpolation [2, 19, 20], denoising [18], super-resolution [11, 31], semantic segmentation [29], and point cloud segmentation [33]. For the deblurring task, Zhang *et al.* [35] proposed pixel-recurrent adaptive convolution for motion deblurring. However, their method requires massive computational cost as the recurrent neural network must run for each pixel. Zhou *et al.* [37] proposed a novel filter adaptive convolution layer for frame alignment and video deblurring. However, handling large motion blur requires predicting large filters, which results in huge computational cost.

## 3. Single Image Defocus Deblurring

In this section, we first introduce the Iterative Adaptive Convolution (IAC) layer, which forms the basis of our IFAN (Sec. 3.1). Then, we present our deblurring network based on IFAN with detailed explanations on each component (Sec. 3.2). Finally, we explain our training strategy exploiting the disparity estimation and reblurring tasks (Sec. 3.3).

### 3.1. Iterative Adaptive Convolution (IAC)

Our IAC is inspired by the filter adaptive convolution (FAC) [37] that forms the basis of previous FANs. For a better understanding of IAC, we first briefly review FAC.

Fig. 1 shows an overview of a FAC layer. A FAC layer takes a feature map $e$ and a map of spatially-varying convolution filters $\mathbf{F}$ as input. $e$ and $\mathbf{F}$ have the same spatial size $h \times w$. The input feature map $e$ has $c$ channels. At each spatial location in $\mathbf{F}$ is a $ck^2$-dim vector representing $c$ convolution filters of size $k \times k$. For each spatial location $(x, y)$, the FAC layer generates $c$ convolution filters from $\mathbf{F}$ by reshaping the vector at $(x, y)$. Then, the layer applies the filters to the features in $e$ centered at $(x, y)$ in the channel-wise manner to generate an output feature map $\hat{e} \in \mathbb{R}^{h \times w \times c}$.

For effectively handling spatially-varying and large defocus blur, it is critical to secure large receptive fields. However, while FAC facilitates spatially-adaptive processing of features, increasing the filter size to cover wider receptive fields results in huge memory consumption and computa-

tional cost. To resolve this limitation, we propose the IAC layer that iteratively applies small-sized separable filters to efficiently enlarge the receptive fields with little computational overhead.

Fig. 2 shows an overview of our IAC layer. Similarly to FAC, IAC takes a feature map $e$ and a map of spatially-varying filters $\mathbf{F}$ as input, whose spatial sizes are the same. At each spatial location in $\mathbf{F}$ is a $Nc(2k+1)$-dim vector representing $N$ sets of filters $\{\mathbf{F}^1, \mathbf{F}^2, \cdots, \mathbf{F}^N\}$. The $n$-th filter set $\mathbf{F}^n$ has two 1-dim filters $\mathbf{f}_1^n$ and $\mathbf{f}_2^n$ of the sizes $k \times 1$ and $1 \times k$, respectively, and one bias vector $\mathbf{b}^n$. $\mathbf{f}_1^n$, $\mathbf{f}_2^n$, and $\mathbf{b}^n$ have $c$ channels. The IAC layer decomposes the vector in $\mathbf{F}$ at each location into filters and bias vectors, and iteratively applies them to $e$ in the channel-wise manner to produce an output feature map $\hat{e}$.

Let $\hat{e}^n$ denote the $n$-th intermediate feature map after applying the $n$-th separable filters and bias, where $\hat{e}^0 = e$ and $\hat{e}^N = \hat{e}$. Then, the IAC layer computes $\hat{e}^n$ for $n \in \{1, \cdots, N\}$ as follows:

$$\hat{e}^n = \text{LReLU}(\hat{e}^{(n-1)} * \mathbf{f}_1^n * \mathbf{f}_2^n + \mathbf{b}^n), \qquad (1)$$

where LReLU is the leaky rectified linear unit [17] and $*$ is the channel-wise convolution operator that performs convolutions in a spatially-adaptive manner.

Separable filters in our IAC layer play a key role in resolving the limitation of the FAC layer. Xu *et al.* [34] showed that a convolutional network with 1-dim filters can successfully approximate a large inverse filter for the deconvolution task. Similarly, our IAC layer secures larger receptive fields at much lower memory and computational costs than the FAC layer by utilizing 1-dim filters, instead of 2-dim convolutions. However, compared to dense 2-dim convolution filters in the FAC layer, our separable filters may not provide enough accuracy for deblurring filters. We handle this problem by iteratively applying separable filters to fully exploit the nonlinear nature of a deep network. Our iterative scheme also enables small-sized separable filters to be used for establishing large receptive fields.

## 3.2. Deblurring Network with IFAN

Fig. 3 shows an overview of our deblurring network based on IFAN. Our network takes a single defocused image $I_B \in \mathbb{R}^{H \times W \times 3}$ as input and produces a deblurred output $I_{BS} \in \mathbb{R}^{H \times W \times 3}$, where $H$ and $W$ are the height and width of an image, respectively. The network is built upon a simple encoder-decoder architecture consisting of a feature extractor, reconstructor, and IFAN module in the middle. The feature extractor extracts defocused features $e_B \in \mathbb{R}^{h \times w \times c_e}$, where $h = \frac{H}{8}$ and $w = \frac{W}{8}$, and feeds them to IFAN. IFAN removes defocus blur in the feature domain by predicting spatially-varying deblurring filters and applying them to $e_B$ using IAC. The deblurred features $e_{BS}$ from IFAN is then passed to the reconstructor, which restores an all-in-focus
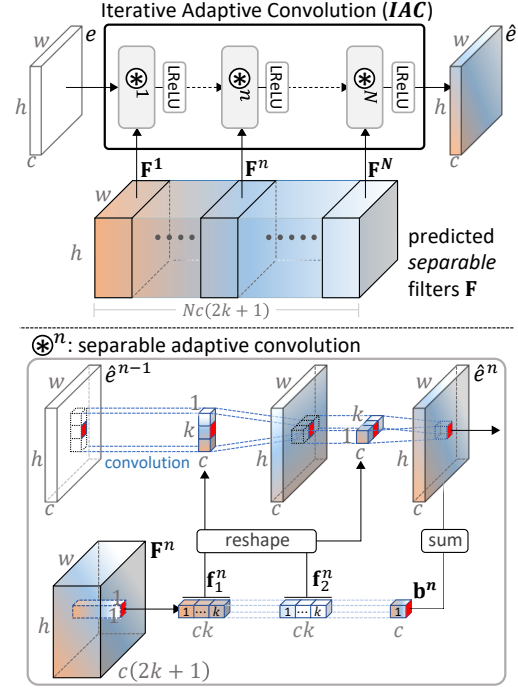


Figure 2. Proposed Iterative Adaptive Convolution (IAC)

image $I_{BS}$. The detailed architecture of each network component can be found in our supplementary material. In the following, we describe IFAN in more detail.

**Iterative Filter Adaptive Network (IFAN)** IFAN takes defocused features $e_B$ as input, and produces deblurred features $e_{BS}$. To produce deblurred features, IFAN takes $I_B$ as additional input, and predicts deblurring filters from $I_B$. IFAN consists of a filter encoder, disparity map estimator, filter predictor, and IAC layer. The filter encoder encodes $I_B$ into $e_{\mathbf{F}} \in \mathbb{R}^{h \times w \times c_e}$, which is then passed to the disparity map estimator. The disparity map estimator is a subnetwork specifically designed to exploit dual-pixel data for effectively training IFAN, which will be discussed in Sec. 3.3. After the disparity map estimator, the filter predictor predicts a deblurring filter map $\mathbf{F}_{deblur} \in \mathbb{R}^{h \times w \times c_{\mathbf{F}_{deblur}}}$, where $c_{\mathbf{F}_{deblur}} = Nc_e(2k+1)$. Finally, the IAC layer transforms the input features $e_B$ using the predicted filters $\mathbf{F}_{deblur}$ to generate deblurred features $e_{BS}$.

## 3.3. Network Training

We train our network using the DPDD dataset with two defocus-specific tasks: defocus disparity estimation and reblurring. In this section, we explain the training data, our training strategy using each task, and our final loss function.

**Dataset** We use dual-pixel images from the DPDD dataset [1] to train our network. A dual-pixel image provides a pair of stereo images with a tiny baseline, whose disparities are proportional to defocus blur magnitudes. The DPDD dataset provides 500 dual-pixel images captured
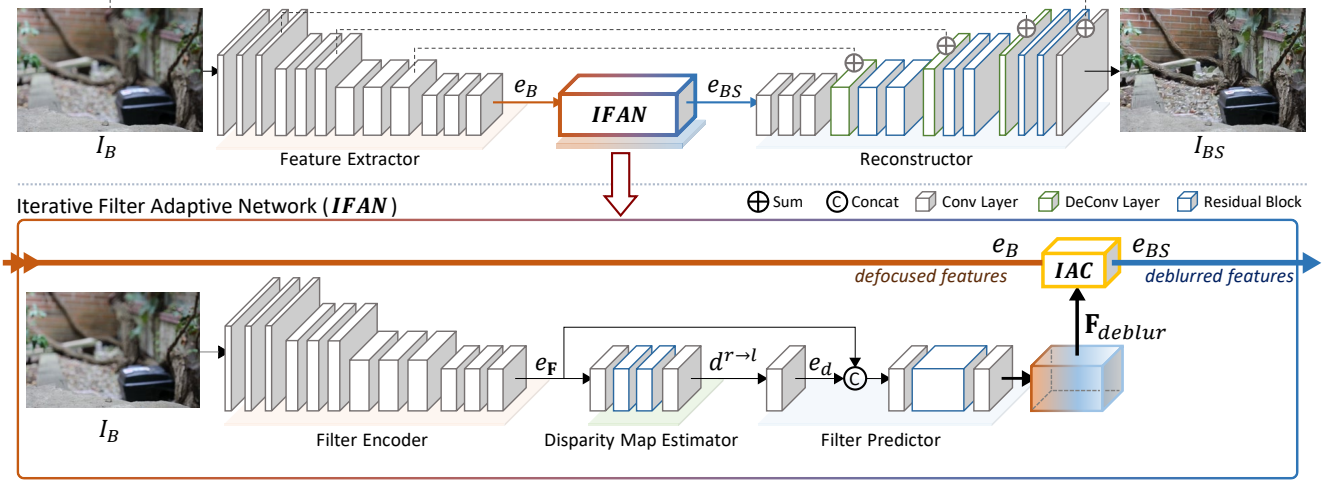
Figure 3. Proposed defocus deblurring network with Iterative Filter Adaptive Network (IFAN).
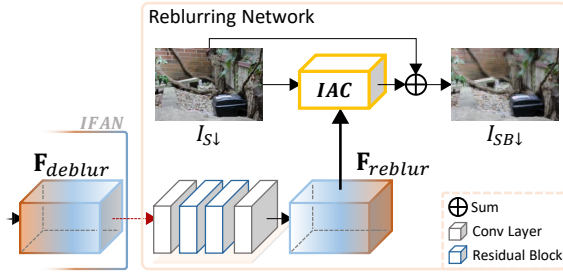


Figure 4. The reblurring network

with a Canon EOS 5D Mark IV. Each dual-pixel image is provided in the form of a pair of left and right stereo images $I_B^l$ and $I_B^r$, respectively. For each dual-pixel image, the dataset also provides a defocused image $I_B$, which is generated by merging $I_B^l$ and $I_B^r$, and its corresponding ground-truth all-in-focus image $I_S$. The 500 dual-pixel images are split into training, validation, and testing sets, each of which contains 350, 74, and 76 scenes, respectively. Refer to [1] for more details on the DPDD dataset.

**Defocus disparity estimation** As the disparities between dual-pixel stereo images are proportional to blur magnitudes, we can train IFAN to learn more accurate defocus blur information by training IFAN to predict the disparity map. To this end, during training, we feed a right image $I_B^r$ from a pair of dual-pixel stereo images to our deblurring network. Then, we train the disparity map estimator to predict a disparity map $d^{r \to l} \in \mathbb{R}^{h \times w}$ between the downsampled left and right stereo images. Specifically, we train the disparity map estimator using a disparity loss defined as:

$$\mathcal{L}_{disp} = MSE(I_{B\downarrow}^{r \to l}, I_{B\downarrow}^l), \quad (2)$$

where $MSE(\cdot)$ is the mean-squared error function. $I_{B\downarrow}^l$ is a left image downsampled by $\frac{1}{8}$. $I_{B\downarrow}^{r \to l}$ is a right image downsampled by $\frac{1}{8}$ and warped by the disparity map $d^{r \to l}$. We use the spatial transformer [10] for warping. By minimizing $\mathcal{L}_{disp}$, both filter encoder and disparity map estimator

are trained to predict an accurate disparity map and eventually accurate defocus magnitudes. Note that we utilize dual-pixel images only for training, and our trained network requires only a single defocused image as its input.

**Reblurring** We train IFAN also using the reblurring task. For the learning of reblurring, we introduce an auxiliary reblurring network. The reblurring network is attached at the end of IFAN and trained to invert deblurring filters $\mathbf{F}_{deblur}$ to reblurring filters $\mathbf{F}_{reblur}$ (Fig. 4). Then, using $\mathbf{F}_{reblur}$, the IAC layer reblurs a downsampled ground-truth image $I_{S\downarrow} \in \mathbb{R}^{h \times w \times 3}$ to reproduce a downsampled version of the defocused input image. For training IFAN as well as the reblurring network, we use a reblurring loss defined as:

$$\mathcal{L}_{reblur} = MSE(I_{SB\downarrow}, I_{B\downarrow}), \quad (3)$$

where $I_{SB\downarrow}$ is a reblurred image obtained from $I_{S\downarrow}$ using $\mathbf{F}_{reblur}$, and $I_{B\downarrow}$ is a downsampled input image. $\mathcal{L}_{reblur}$ induces IFAN to predict $\mathbf{F}_{deblur}$ containing valid information about blur shapes and sizes needed for accurate reblurring. Such information eventually improves the performance of deblurring filters used for the final defocus deblurring. Note that we utilize the reblurring network only for training.

**Loss functions** In addition to the disparity and reblurring losses, we use a deblurring loss, which is defined as:

$$\mathcal{L}_{deblur} = MSE(I_{BS}, I_S). \quad (4)$$

Our total loss function to train our network is then defined as $\mathcal{L}_{total} = \mathcal{L}_{deblur} + \mathcal{L}_{disp} + \mathcal{L}_{reblur}$. Each loss term affects different parts of our network. While $\mathcal{L}_{deblur}$ trains the feature extractor, IFAN, and reconstructor, $\mathcal{L}_{disp}$ trains only the filter encoder and disparity map estimator in IFAN. $\mathcal{L}_{reblur}$ trains both IFAN and reblurring network. Note that we use dual-pixel stereo images $(I_B^l, I_B^r)$ only for training. Both $\mathcal{L}_{deblur}$ and $\mathcal{L}_{reblur}$ utilize $I_B$ while $\mathcal{L}_{disp}$ utilizes $(I_B^l, I_B^r)$. In this way, we can fully utilize the DPDD dataset for training our network.

| IFAN | | RBN | Evaluations on the DPDD Dataset [1] | | | | Computational Costs | |
|---|---|---|---|---|---|---|---|---|
| FP + IAC | DME | | PSNR↑ | SSIM↑ | MAE$_{(\times10^{-1})}$↓ | LPIPS↓ | Params (M) | MACs (B) |
| | | | 24.88 | 0.753 | 0.416 | 0.289 | 10.58 | 364.3 |
| | ✓ | | 24.97 | 0.761 | 0.412 | 0.280 | | |
| ✓ | | | 25.07 | 0.765 | 0.406 | 0.271 | 10.48 | 362.9 |
| ✓ | ✓ | | 25.18 | 0.780 | 0.403 | 0.233 | | |
| ✓ | | ✓ | 25.28 | 0.780 | 0.400 | 0.245 | | |
| ✓ | ✓ | ✓ | **25.37** | **0.789** | **0.394** | **0.217** | | |

Table 1. An ablation study. FP, DME, and RBN indicate the filter predictor, disparity map estimator, and reblurring network, respectively. The first row corresponds to the baseline model. For fair evaluation, to obtain the baseline model, the components of our model are replaced by conventional convolution layers and residual blocks with similar model parameter numbers and computational costs.
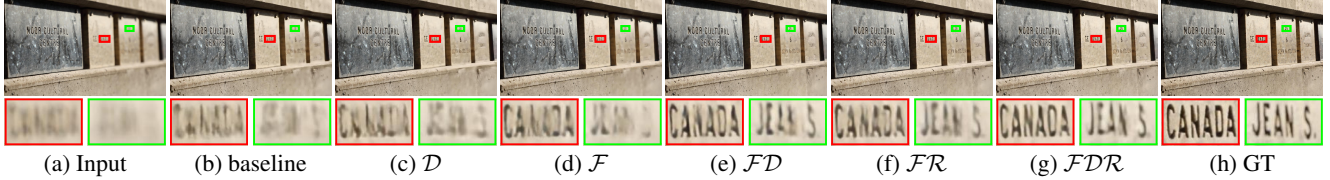


Figure 5. Qualitative results of an ablation study on the DPDD dataset [1]. The first and last columns show a defocused input image and its ground-truth all-in-focus image, respectively. Between the columns, the letters in each sub-caption indicate combination of components (refer Table 1). $\mathcal{F}$ means the filter predictor and the IAC layer, $\mathcal{D}$ means the disparity map estimator, and $\mathcal{R}$ means the reblurring network. The baseline implies a model without $\mathcal{F}$, $\mathcal{D}$ and $\mathcal{R}$. Images in the red and green box are zoomed-in cropped patches.

(a) Input  (b) baseline  (c) $\mathcal{D}$  (d) $\mathcal{F}$  (e) $\mathcal{FD}$  (f) $\mathcal{FR}$  (g) $\mathcal{FDR}$  (h) GT

## 4. Experiments

We implemented our models using PyTorch [23]. We use rectified-Adam [16] with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and weight decay rate 0.01 for training our network. We use gradient norm clipping with the value empirically set to 0.5. The network is trained for 600k iterations with an initial learning rate of $1.0 \times 10^{-4}$, which is step-decayed to half at the 500k-th and 550k-th iterations, which are experimentally chosen. We set the number of filters $N = 17$ for $\mathbf{F}_{deblur}$ and $\mathbf{F}_{reblur}$. For each iteration, we randomly sample a batch of images from the DPDD training set. We use a batch size of 8, and randomly augment images with Gaussian noise, gray-scale image conversion, and scaling, then crop them to $256 \times 256$.

For the evaluation of defocus deblurring performance, we measure the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [32], Mean Absolute Error (MAE), and Learned Perceptual Image Patch Similarity (LPIPS) [36] between deblurred results and their corresponding ground-truth images. We evaluate our models and previous ones on a PC with an NVIDIA GeForce RTX 3090 GPU.

### 4.1. Ablation Study

To analyze the effect of each component of our model, we conduct an ablation study (Table 1 and Fig. 5). All models in the ablation study are trained under the same conditions (*e.g.*, optimizer, batch size, learning rate, random seeds, etc.). For the validation, we compare a stripped-down baseline model and its five variants. For the baseline model, we restructure the key modules specifically designed for our network using conventional convolution layers and resid-

ual blocks. Specifically, we change the channel sizes of the last layers of the disparity map estimator and filter predictor blocks to $c_e$ so that they predict conventional feature maps. We also replace the IAC layer with residual blocks that take a concatenation of $e_B$ and $\mathbf{F}_{deblur}$ as input, and detach the reblurring network. The baseline model is trained with neither $\mathcal{L}_{disp}$ nor $\mathcal{L}_{reblur}$ but only $\mathcal{L}_{deblur}$. For the other variants, we recover combinations of the restructured components one by one from the baseline model. For the variants with the disparity map estimator, we train them also with $\mathcal{L}_{disp}$. Similarly, for the variants with the reblurring network, we train them also with $\mathcal{L}_{reblur}$. The detailed architectures of all the models used in the ablation study can be found in the supplementary material.

**Explicit deblurring filter prediction** We first verify the effect of the filter prediction scheme implemented using the filter predictor and IAC layer. Table 1 shows that introducing the filter predictor and IAC layer to the baseline model increases the deblurring performance (the first and third rows in the table), confirming the advantage of *explicit* pixel-wise filter prediction in flexible handling of spatially-varying and large defocus blur. In addition, compared to the gain (PSNR: 0.36% and LPIPS: 3.21%) obtained when the disparity map estimator is embedded in the baseline model (the second row in the table), there is more performance gain (PSNR: 0.44% and LPIPS: 16.31%) when the disparity map estimator is added to a model with the filter predictor and IAC layer (the fourth row in the table). This observation validates that explicit utilization of deblurring filters has more potential in absorbing extra defocus blur-specific supervision provided by dual-pixel images. Fig. 5 shows a

| Model | Evaluations the DPDD Dataset [1] | | | | Computational Costs | | |
|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | MAE$_{(\times 10^{-1})}$↓ | LPIPS↓ | Params (M) | MACs (B) | Time (Sec) |
| Input | 23.89 | 0.725 | 0.471 | 0.349 | - | - | - |
| JNB [28] | 23.69 | 0.707 | 0.480 | 0.442 | - | - | 105.8 |
| EBDB [12] | 23.94 | 0.723 | 0.468 | 0.402 | - | - | 96.58 |
| DMENet [14] | 23.90 | 0.720 | 0.470 | 0.410 | 26.94 | 1172.5 | 77.70 |
| DPDNet$_S$ [1] | 24.03 | 0.735 | 0.461 | 0.279 | 35.25 | 989.8 | 0.462 |
| DPDNet$_D$ [1] | 25.23 | 0.787 | 0.401 | 0.224 | 35.25 | 991.4 | 0.474 |
| Ours | **25.37** | **0.789** | **0.394** | **0.217** | **10.48** | **362.9** | **0.014** |

Table 2. Quantitative comparison with previous defocus deblurring methods. All the methods are evaluated using the code provided by the authors. JNB and EBDB are not deep learning-based methods, so their parameter numbers and MACs are not available. DPDNet$_D$ [1] takes not a single defocused image but dual-pixel stereo images as input at test time. All the other methods, including ours, take a single defocused image as input at test time.



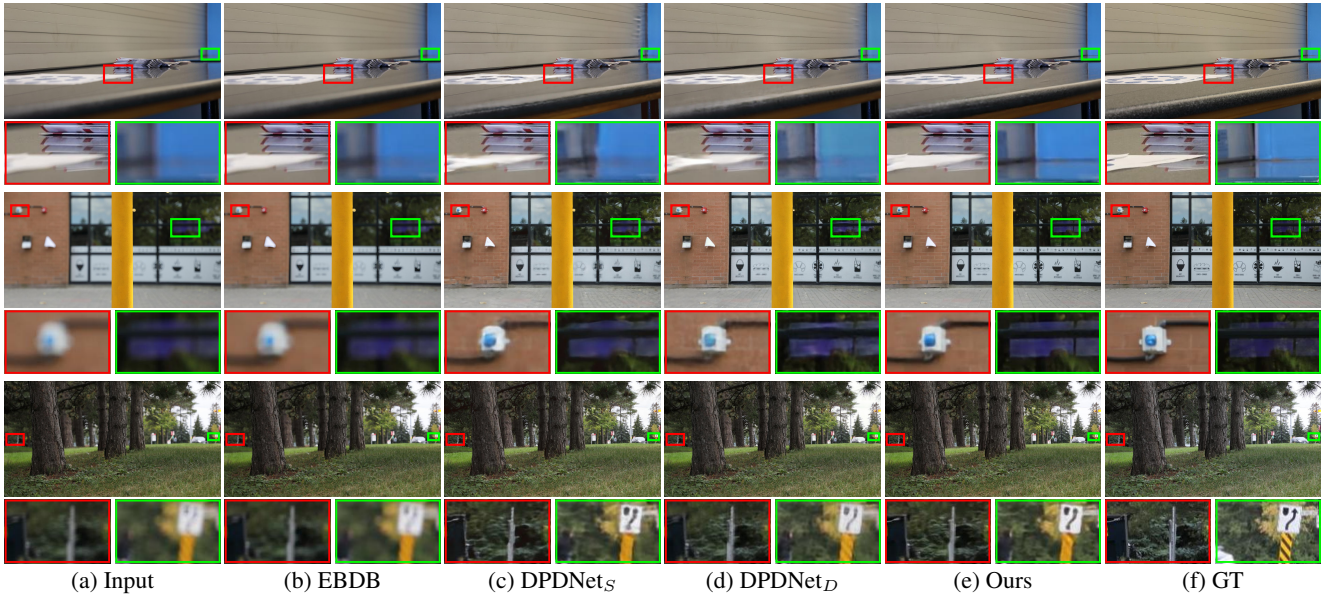(a) Input (b) EBDB (c) DPDNet$_S$ (d) DPDNet$_D$ (e) Ours (f) GT

Figure 6. Qualitative comparison on the DPDD dataset [1]. The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Note that DPDNet$_D$ requires a pair of dual-pixel stereo images as input, while other methods, including ours, require only a single image at test time. Refer to the supplementary material for more results.

qualitative comparison. As shown in the figure, the filter predictor and IAC layer substantially enhance the deblurring quality ((b) and (c) vs. (d) and (e) in the figure).

**Disparity map estimation and reblurring** We analyze the influence of the disparity map estimator and reblurring network. Specifically, we compare the combinations (filter predictor + IAC + disparity map estimator) and (filter predictor + IAC + reblurring network). Table 1 show that the model with the disparity map estimator performs better than the model with the reblurring network in recovering textures (lower LPIPS), as the disparity map estimator helps more accurately estimate per-pixel blur amounts (Fig. 5e and f). On the other hand, the model with the reblurring network better restores overall image contents (higher PSNR), as the reblurring network guides deblurring filters to contain information about blur shapes and blur amounts.

We can also observe that the model with both disparity map estimator and reblurring network achieves the best performance in every measure (Fig. 5g). This shows that the disparity map estimator and reblurring network have synergistic effects, complementing each other.

### 4.2. Comparison with Previous Methods

In this section, we compare our method with previous defocus map-based approaches as well as recent end-to-end learning-based approaches: Just Noticeable Blur estimation (JNB) [28], Edge-Based Defocus Blur estimation (EBDB) [12], Defocus Map Estimation Network (DMENet) [14], DPDNet$_S$ and DPDNet$_D$ [1]. Among these, JNB, EBDB, and DMENet are defocus map-based approaches that first estimate a defocus map and perform non-blind deconvolution. DPDNet$_S$ and DPDNet$_D$ are end-to-end learning-

based approaches that directly restore all-in-focus images. They share the same network architecture, but DPDNet$_S$ takes a single defocused image as input while DPDNet$_D$ takes a pair of dual-pixel stereo images.

For all the previous methods, we use the code (and model weights for learning-based methods, DMENet and DPDNet) provided by the authors. For JNB, EBDB, and DMENet, we used the deconvolution method [13] to obtain all-in-focus images using the estimated defocus maps. For DPDNet$_S$, we retrain the network on 8-bit images with the training code provided by the author, as the authors provide only a model trained with 16-bit images (refer to the supplementary material for the evaluation of our model trained on 16-bit images). We measure computational costs in terms of the number of network parameters, number of multiply-accumulate operations (MACs) computed on a $1280 \times 720$ image, and average computation time computed on test images. For JNB and EBDB, which are not learning-based methods, we measure only their computation times.

We compare the methods on the DPDD test set [1]. Table 2 shows a quantitative comparison. The previous defocus map-based methods show poor performance on the real-world blurred images in the DPDD test set, which is even lower than the input defocused images due to their restrictive blur models. On the other hand, the recent end-to-end approaches, DPDNet$_D$ and DPDNet$_S$, achieve higher quality compared to the previous methods. Our model outperforms DPDNet$_S$ by a significant gap with a smaller computational cost. Moreover, although our model uses a single defocused image, it outperforms DPDNet$_D$ as well, proving the effectiveness of our approach.

Fig. 6 shows a qualitative comparison. Due to their inaccurate defocus maps and restricted blur models, the results of the defocus map-based methods have a large amount of remaining blur (Fig. 6b). DPDNet$_S$ and DPDNet$_D$ produce better results than the previous ones, however, tend to produce artifacts and remaining blur (Fig. 6c and 6d). On the other hand, our method shows more accurate deblurring results (Fig. 6e), even with a single defocused input. Especially, compared to DPDNet$_D$, Fig. 6 show that our method better handles spatially-varying blur (the first row), large blur (the second row), and image structures as well as textures (the third row). More qualitative results are in the supplementary material.

### 4.2.1 Generalization Ability

As our method is trained using the DPDD training set, which is captured by a specific camera (Canon EOS 5D Mark IV), one question naturally follows how well the model generalizes to other images from different cameras. To answer this question, we evaluate the performance of our approach on other test sets.

| Model | PSNR↑ | SSIM↑ | MAE$_{(\times 10^{-1})}$↓ | LPIPS↓ |
|---|---|---|---|---|
| Input | 22.33 | 0.633 | 0.513 | 0.524 |
| JNB [28] | 22.36 | 0.635 | 0.511 | 0.601 |
| EBDB [12] | 22.38 | 0.638 | 0.509 | 0.594 |
| DMENet [14] | 22.41 | 0.639 | 0.508 | 0.597 |
| DPDNet$_S$ [1] | 22.67 | 0.666 | 0.506 | 0.420 |
| Ours | **24.71** | **0.748** | **0.407** | **0.306** |

Table 3. Quantitative evaluation on the RealDOF test set.



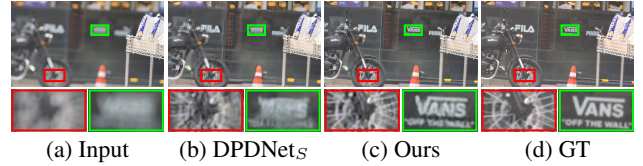(a) Input    (b) DPDNet$_S$    (c) Ours    (d) GT

Figure 7. Qualitative comparison on the RealDOF test set. From left to right: a defocused input image, deblurred results of DPDNet$_S$ [1] and our method, and a ground-truth image.

**RealDOF test set** To quantitatively measure the performance of our method on real-world defocus blur images, we prepare a new dataset named *Real Depth of Field (RealDOF)* test set. RealDOF consists of 50 scenes. For each scene, the dataset provides a pair of a defocused image and its corresponding all-in-focus image. To capture image pairs of the same scene with different depth-of-fields, we built a dual-camera system with a beam splitter as described in [25]. Specifically, our system is equipped with two cameras attached to the vertical rig with a beam splitter. We used two Sony a7R IV cameras, which do not support dual pixels, with Sony 135mm F1.8 lenses. The system is also equipped with a multi-camera trigger to synchronize the camera shutters to capture images simultaneously. The captured images are post-processed for geometric and photometric alignments, similarly to [25]. More details about the RealDOF test set are in the supplementary material.

Table 3 shows a quantitative comparison on the RealDOF test set. The table shows that our model clearly improves the image quality, showing that the model can generalize well to images from other cameras. Moreover, our model significantly outperforms the previous state-of-the-art single image deblurring method, DPDNet$_S$, by more than 2 dB in terms of PSNR. Fig. 7 qualitatively compares our method and DPDNet$_S$. While the result of DPDNet$_S$ contains some amount of remaining blur, ours looks much sharper with no remaining blur.

**CUHK blur detection dataset** The CUHK blur detection dataset [27] provides 704 defocused images collected from the internet without ground-truth all-in-focus images. Fig. 8 shows a qualitative comparison between DPDNet$_S$ and ours on the CUHK dataset. The result shows that our method removes defocus blur and restores fine details more successfully than DPDNet$_S$. Refer to the supplementary material for more qualitative results.

| $N$(RF) | Deblurring Results | | | | Params (M) | MACs (B) |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | MAE$_{(\times10^{-1})}$↓ | LPIPS↓ | | |
| 8 (17) | 25.19 | 0.777 | 0.404 | 0.246 | **9.44** | **347.9** |
| 17 (35) | 25.37 | 0.789 | 0.394 | 0.217 | 10.48 | 362.9 |
| 26 (53) | 25.39 | 0.788 | 0.393 | 0.215 | 11.52 | 377.9 |
| 35 (71) | 25.42 | 0.789 | 0.391 | 0.213 | 12.56 | 392.8 |
| 44 (89) | **25.45** | **0.792** | **0.389** | **0.206** | 13.60 | 407.8 |

Table 4. Deblurring performance and computational cost with respect to the number of deblurring filters $N$ evaluated on the DPDD dataset [1]. RF denotes the receptive field size.



(a) Input     (b) DPDNet$_S$     (c) Ours

Figure 8. Qualitative comparison on the CUHK blur detection dataset [27]. From left to right: a defocused input image, deblurred results of DPDNet$_S$ [1] and our method.

**Pixel dual-pixel test set**  The DPDD dataset [1] provide an additional test set consisting of dual-pixel defocused images captured by a Google Pixel 4 smartphone camera. We refer the readers to the supplementary material for the qualitative evaluation on the Pixel dual-pixel test set.

### 4.3. Analysis on IFAN

In this section, we further investigate the effect of different components of IFAN. We first analyze the effect of the number of separable filters $N$ in $\mathbf{F}_{deblur}$. Then, we evaluate the effect of IAC compared to FAC [37].

**Number of separable filters $N$**  A larger number of separable filters in $\mathbf{F}_{deblur}$ leads IFAN to establish larger receptive fields and more accurate deblurring filters. Consequently, when $N$ is larger, we can handle large defocus blur more accurately. Table 4 compares the performances with different values of $N$, where the deblurring quality increases with $N$. Based on the result, we choose $N = 17$ for our final model, as the improvement is small for $N > 17$.

**IAC vs. FAC**  Finally, we analyze the effectiveness of the proposed IAC compared to FAC [37]. To compare IAC and FAC, we replace the IAC layers in both IFAN and reblurring network in our final model with FAC layers. For the FAC layers, we use $k = 11$ to match the computational cost to that of our final model for fairness. Table 5 and Fig. 9 respectively show quantitative and qualitative comparisons between IAC and FAC. The comparisons show that IAC outperforms FAC even with fewer parameters and operations, as IAC is better in handling large defocus blur by cov-

| Module | Deblurring Results | | | | Params (M) | MACs (B) |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | MAE$_{(\times10^{-1})}$↓ | LPIPS↓ | | |
| FAC | 25.18 | 0.778 | 0.406 | 0.249 | 10.51 | 363.4 |
| IAC | **25.37** | **0.789** | **0.394** | **0.217** | **10.48** | **362.9** |

Table 5. Quantitative comparison between the FAC [37] and IAC layers evaluated on the DPDD dataset [1]. IAC indicates our model with IAC layers, while FAC indicates a variant of our model whose IAC layers are replaced with FAC layers. We set the filter size to $11 \times 11$ for the FAC layers for fairness in computational costs.



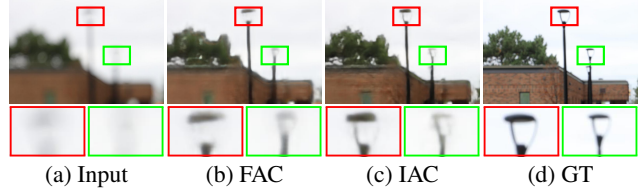(a) Input    (b) FAC    (c) IAC    (d) GT

Figure 9. Qualitative comparison between the FAC and IAC layers evaluated on the DPDD dataset [1]. FAC in (b) means our final model whose IAC layers are replaced with FAC layers. IAC in (c) means our final model with IAC layers. The input blurred image has large defocus blur, so details in the red and green boxes are not visible. Our final model with IAC shows better restored details compared to the model with FAC.

ering a much larger receptive field ($35 \times 35$) on defocused features than the receptive field ($11 \times 11$) of FAC.

## 5. Conclusion

This paper proposed a single image defocus deblurring framework based on our novel Iterative Filter Adaptive Network (IFAN). IFAN flexibly handles spatially-varying defocus blur by predicting per-pixel separable deblurring filters. For efficient and effective handling of a large blur, we proposed Iterative Adaptive Convolution (IAC) that iteratively applies separable filters on features. In addition, IFAN learns to estimate defocus blur from a single image more accurately through the learning of disparity map estimation and reblurring. In the experiments, we verified the effect of each component in our model, and showed that our method achieves state-of-the-art performance.

The proposed network is still limited in handling significantly large defocus blur (*e.g.*, Fig. 9c has remaining blur). Our network works best with a typical isotropic defocus blur, but may not properly handle blur with irregular shape (*e.g.*, swirly bokeh) or strong highlight (*i.e.*, glitter bokeh). Refer to the supplementary material for the failure cases. We plan to extend our RealDOF test set by adding more image pairs of diverse blur types captured with various cameras and lenses.

# References

[1] Abdullah Abuolaim and Michael Brown. Defocus deblurring using dual-pixel data. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[2] Bert Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NIPS*, 2016. 2

[3] Jiale Cao, Hisham Cholakkal, Rao Anwer, Fahad Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. In *CVPR*, 2020. 1

[4] Huaijin Chen, Jinwei Gu, Orazio Gallo, Ming-Yu Liu, Ashok Veeraraghavan, and Jan Kautz. Reblur2deblur: Deblurring videos via self-supervised learning. In *ICCP*, 2018. 2

[5] Sunghyun Cho and Seungyong Lee. Convergence analysis of MAP based blur kernel estimation. In *ICCV*, 2017. 1, 2

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 1

[7] Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Susstrunk. Non-parametric blur map regression for depth of field extension. *IEEE TIP*, 25(4):1660–1673, 2016. 1, 2

[8] D. A. Fish, A. M. Brinicombe, E. R. Pike, and J. G. Walker. Blind deconvolution by means of the richardson-lucy algorithm. *JOSA A*, 12(1):58–65, 1995. 1, 2

[9] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan Barron. Learning single camera depth estimation using dual-pixels. In *ICCV*, 2019. 1

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 4

[11] Younghyun Jo, Seoung Oh, Jaeyeon Kang, and Seon Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 2

[12] Ali Karaali and Claudio Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE TIP*, 27(3):1126–1137, 2018. 1, 2, 6, 7

[13] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 2009. 1, 2, 7

[14] Junyong Lee, Sungkill Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *CVPR*, 2019. 1, 2, 6, 7

[15] Anat Levin, Robert Fergus, Frédo Durand, and William Freeman. Image and depth from a conventional camera with a coded aperture. In *SIGGRAPH*, 2007. 1, 2

[16] Liyuan Liu, Haoming Jiang, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *ICLR*, 2020. 5

[17] Andrew L. Mass, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 3

[18] Ben Mildenhall, Jonathan Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *CVPR*, 2018. 2

[19] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017. 2

[20] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017. 2

[21] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 1

[22] Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and Inso Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *CVPR*, 2017. 1, 2

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPSW*, 2017. 5

[24] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael Brown. Modeling defocus-disparity in dual-pixel sensors. In *ICCP*, 2020. 1

[25] Jaesung Rim, Lee Haeyun, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020. 7

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[27] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *CVPR*, 2014. 2, 7, 8

[28] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015. 1, 2, 6, 7

[29] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *CVPR*, 2019. 2

[30] Li Wang, Dong Li, Yousong Zhu, Lu Tian, and Yi Shan. Dual super-resolution learning for semantic segmentation. In *CVPR*, 2020. 1

[31] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 2

[32] Zhou Wang, Alan Bovik, Hamid R. Sheikh, and Eero Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[33] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *ECCV*, 2020. 2

[34] Li Xu, Jimmy S. J. Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014. 3

[35] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *CVPR*, 2018. 1, 2

[36] Richard Zhang, Phillip Isola, Alexei Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5

[37] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Wangmeng Zuo, Haozhe Xie, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *ICCV*, 2019. 1, 2, 8