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 naive 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 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 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 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 $F$ as input. $e$ and $F$ have the same spatial size $h \times w$. The input feature map $e$ has $c$ channels. At each spatial location in $F$ 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 $F$ 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 $\hat{e} \in R^{h \times w \times c}$.

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 \( F \) as input, whose spatial sizes are the same. At each spatial location in \( F \) is a \( Nc(2k+1) \)-dim vector representing \( N \) sets of filters \( \{F^1, F^2, \ldots, F^N\} \). The \( n \)-th filter set \( F^n \) has two 1-dim filters \( f^1_n \) and \( f^2_n \) of the sizes \( k \times 1 \) and \( 1 \times k \), respectively, and one bias vector \( b^n \). \( f^1_n \), \( f^2_n \), and \( b^n \) have \( c \) channels. The IAC layer decomposes the vector in \( F \) 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 \( \hat{e}^{n-1} = \hat{e} \). Then, the IAC layer computes \( \hat{e}^{n} \) for \( n \in \{1, \ldots, N\} \) as follows:

\[
\hat{e}^{n} = \text{LReLU}(\hat{e}^{(n-1)} \ast f^1_n \ast f^2_n + b^n), \tag{1}
\]

where LReLU is the leaky rectified linear unit [17] and \( \ast \) 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 \( h = \frac{H}{4} \) and \( w = \frac{W}{4} \), 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 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 \( e_F \in \mathbb{R}^{h \times w \times c_e} \), which is then passed to the disparity map estimator. The disparity map estimator is a sub-network 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 \( F_{deblur} \in \mathbb{R}^{h \times w \times c_{deblur}} \), where \( c_{deblur} = Nc_e(2k + 1) \). Finally, the IAC layer transforms the input features \( e_B \) using the predicted filters \( F_{deblur} \) 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 deblurring. 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...
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 $d^{r \rightarrow l} \in \mathbb{R}^{h \times w}$ between the downsampled left and right stereo images. Specifically, we train the disparity map estimator using a disparity loss defined as:

$$L_{\text{disp}} = \text{MSE}(I_B^{r \rightarrow l} , I_B^l),$$  

where $\text{MSE}(\cdot)$ is the mean-squared error function. $I_B^{r \rightarrow l}$ is a left image downsampled by $\frac{1}{8}$. $I_B^l$ is a right image downsampled by $\frac{1}{8}$ and warped by the disparity map $d^{r \rightarrow l}$. We use the spatial transformer [10] for warping. By minimizing $L_{\text{disp}}$, 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 $F_{\text{deblur}}$ to reblurring filters $F_{\text{reblur}}$ (Fig. 4). Then, using $F_{\text{reblur}}$, the IAC layer reblurs a downscaled ground-truth image $I_{SB} \in \mathbb{R}^{h \times w \times 3}$ 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:

$$L_{\text{reblur}} = \text{MSE}(I_{SB} | I_B),$$  

where $I_{SB}$ is a reblurred image obtained from $I_B$ using $F_{\text{reblur}}$, and $I_B$ is a downscaled input image. $L_{\text{reblur}}$ induces IFAN to predict $F_{\text{reblur}}$ 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:

$$L_{\text{deblur}} = \text{MSE}(I_B, I_S).$$  

Our total loss function to train our network is then defined as $L_{\text{total}} = L_{\text{deblur}} + L_{\text{disp}} + L_{\text{reblur}}$. Each loss term affects different parts of our network. While $L_{\text{deblur}}$ trains the feature extractor, IFAN, and reconstructor, $L_{\text{disp}}$ trains only the filter encoder and disparity map estimator in IFAN. $L_{\text{reblur}}$ trains both IFAN and reblurring network. Note that we use dual-pixel stereo images $(I_B^l, I_B^r)$ only for training. Both $L_{\text{deblur}}$ and $L_{\text{reblur}}$ utilize $I_B$, while $L_{\text{disp}}$ utilizes $(I_B^l, I_B^r)$. In this way, we can fully utilize the DPDD dataset for training our network.
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). F means the filter predictor and the IAC layer, D means the disparity map estimator, and R means the reblurring network. The baseline implies a model without F, D and R. Images in the red and green box are zoomed-in cropped patches.

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 \( 1 \times 10^{-4} \), 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 \( \mathbf{F}_{\text{deblur}} \) and \( \mathbf{F}_{\text{reblur}} \). 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 256×256.

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 residual 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 \( \mathbf{F}_{\text{deblur}} \) as input, and detach the reblurring network. The baseline model is trained with neither \( \mathcal{L}_{\text{disp}} \) nor \( \mathcal{L}_{\text{reblur}} \) but only \( \mathcal{L}_{\text{deblur}} \). 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 \( \mathcal{L}_{\text{disp}} \). Similarly, for the variants with the reblurring network, we train them also with \( \mathcal{L}_{\text{reblur}} \). 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
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.

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.
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, DME1Net and DPDNet provided by the authors. For JNB, EBDB, and DME1Net, 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 × 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  |
| DME1Net [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.

Figure 7. Qualitative comparison on the RealDOF test set. From left to right: a defocused input image, deblurred results of DPDNet$_S$ 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 qualitative comparison on the RealDOF test set. The table shows that our model clearly improves the image quality, allowing that the model can provide 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.
## Deblurring Results

| Deblurring Results | Params MACs (M) | (B) |
|---|---|---|
| **N(RF)** | **PSNR↑** | **SSIM↑** | **MAE(×10⁻¹)↓** | **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.

## 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 \(F_{deblur}\). Then, we evaluate the effect of IAC compared to FAC [37].

### Number of separable filters \(N\)

A larger number of separable filters in \(F_{deblur}\) 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 covering 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.

### Acknowledgements

This work was supported by the Ministry of Science and ICT, Korea, through IITP grants (SW Star Lab, 2015-0-00174; Artificial Intelligence Graduate School Program (POSTECH), 2019-0-01906) and NRF grants (2018R1A5A1060031; 2020R1C1C1014863).
1. RealDOF Test Set

We present the Real Depth of Field (RealDOF) test set for quantitative and qualitative evaluations of single image defocus deblurring. Our RealDOF test set contains image pairs, each of which consists of a defocused image and its corresponding all-in-focus image that have been concurrently captured for the same scene.

To simultaneously capture such pairs of images, we built a dual-camera system (Fig. 1), where two cameras are vertically and horizontally aligned to each other on a vertical camera rig. The rig is designed to physically align the cameras as precisely as possible to ensure accurate alignment between the captured images. On the rig, we install a beam splitter between the cameras whose lenses are facing toward the beam splitter. The rig is encased with an optical enclosure that blocks light coming from outside of the viewing direction. In the system, we use the same models for the cameras (Sony A7R IV) and lenses (Sony 135mm F1.8). The cameras are connected with a multi-camera trigger for simultaneous capturing of images. We set the cameras with the minimum shutter speed of $1/125$ seconds to avoid motion blur. ISO is adjusted accordingly for the images to pick up the same exposure.

Using the dual-camera system, we first capture defocused and all-in-focus images (namely, target image pairs). For a target image pair, one camera captures a defocused image with a wide aperture (F1.8 – 5.6), and the other camera obtains an all-in-focused image with a narrow aperture (F16). Images are captured in a 14-bit raw (Sony ARW) with the resolution of $9504 \times 6344$. The captured images are then processed to an sRGB using Adobe Lightroom and encoded with a lossless 16-bit TIFF format. Then, the encoded images are resized to $2376 \times 1586$, and geometric and photometric alignments are performed. As it is ambiguous to geometrically align blurry and sharp frames, we additionally capture all-in-focus image pairs (namely, reference image pairs) of the same scene, motivated by [4]. For the reference image pairs, we set both cameras with the same aperture (F16).

Although the vertical rig used for our system is designed to accurately align the cameras, physical misalignment may still exist. Besides, the cameras may move slightly over time due to shakes. To handle such physical misalignments, we apply geometric alignments to captured images. We first compute a homography matrix from the reference image pair. As in [4], we use the enhanced correlation coefficients method [2], which is robust to photometric misalignment. Then, the matrix is applied to the corresponding target image pair, where the all-in-focused image is geometrically aligned to the defocused image using the matrix.

We use the same models for the two cameras and their lenses, but captured images still may exhibit exposure differences. We address this issue with photometric alignment based on a linear model as in [4]. Specifically, we compute linear photometric parameters from a target image pair, and then apply the parameters to the all-in-focus image to match its exposure to that of the defocused image. The final RealDOF test set contains geometrically and photometrically aligned target pairs of defocused and all-in-focused images.
2. Failure Cases

Our network works best with typical isotropic defocus blur, and may not properly handle blur with irregular shape (e.g. swirly bokeh as in the first row of Fig. 2) or strong highlight (i.e. glitter bokeh as in the second row).

3. Our Model with Different Input Types

16-bit images Our final model is trained on 8-bit images, as most standard encodings still rely on 8-bit images. Nonetheless, we also show the capability of our model in handling high bit-depth images, as the final model of DPDNet is targeted for 16-bit images. Table 1 shows a quantitative comparison on the DPDD dataset between our model and DPDNets that are trained and tested for 16-bit images. Our model outperforms DPDNets for all the deblurring metrics.

Dual-pixel images The deblurring performance of our model further increases if we explicitly feed dual-pixel images. Table 2 and Figs. 3, 4, and 5 show quantitative and qualitative comparisons of our model fed with dual-pixel images (OursD). OursD has the same architecture of our final model taking a single image, except that the filter encoder in IFAN is fed with dual-pixel stereo images (I_d and I_p) that are concatenated along the channel dimension. In the figures, Ours_dual shows better performance in handling spatially-varying (the red and green boxes at different focal planes in the figures) and large (green boxes) defocus blur.

4. Effect of Noise Augmentation Level

Table 3 shows the effect of the noise level used to augment training images. For training each model in the table, defocused images are randomly augmented with Gaussian noise, controlled by a random standard deviation within a range [0, σ]. We can infer from the table that compared to a model trained with a low noise level, a model trained with a higher noise level is better in restoring overall image contents (higher PSNR), but worse in recovering textures (higher LPIPS).

5. Additional Qualitative Results

We show qualitative results on the DPDD test set [1] (Figs. 6, 7, and 8). For the qualitative results of real-world defocused images in different datasets, we show the results of the proposed RealDOF test set (Figs. 9 and 10), Pixel dual-pixel test set [1] (Figs. 11 and 12), and the CUHK dataset [5] (Figs. 13 and 14).

For the CUHK dataset, for readers to visually understand the capability of our method in handling spatially-varying and large blur, we visualize Gaussian point-spread-function (PSF) for each sampled coordinate of defocused images. Each PSF is generated using σ at the corresponding coordinates of a defocus map, which is computed by [3] and contains per-pixel standard deviations for a Gaussian kernel. Note that each PSF is displayed in a 31 × 31 grid and normalized for the visualization.

6. Network Architectures

The detailed network architectures of the models used for the ablation study (Sec. 4.1. of the main paper) are shown in Tables 4, 5, 6, 7, and Fig. 15.

For the tables, type, input, act, k, c, s, p and N in columns denote the type of a layer, input for the layer, activation function, kernel size, out-channels, stride, padding, and repeating number, respectively. For leaky-ReLU (lrelu), we use 0.1 for its negative slope. For the layer types, we have conv, dconv, identity, cat, and sum, which denote convolution, deconvolution, identity, concatenation, and element-wise summation layers, respectively.
Figure 3. Qualitative comparison of dual-pixel image-based models 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 4. Qualitative comparison of dual-pixel image-based models 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 5. **Qualitative comparison of dual-pixel image-based models 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 6. Additional qualitative results 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 7. Additional qualitative results 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 8. Additional qualitative results 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 9. **Additional qualitative comparison results on the proposed RealDOF test set.** 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 10. **Additional qualitative comparison results on the proposed RealDOF test set.** 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. Images in the red and green boxes are zoomed-in cropped patches.
Figure 11. Qualitative comparison on the Pixel Dual-Pixel dataset [1]. The first columns show defocused input images, and for the other columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.
Figure 12. Qualitative comparison on the Pixel Dual-Pixel dataset [1]. The first columns show defocused input images, and for the other columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.
Figure 13. **Qualitative comparison on the CUHK dataset** [5]. The first and last columns show defocused input images and their per-pixel Gaussian PSFs visualized on sampled coordinates, respectively. Between the columns, we show deblurring results of different methods. Green and red boxes highlight the regions with large defocus blur and their deblurred results, respectively.
Figure 14. Qualitative comparison on the CUHK dataset [5]. The first and last columns show defocused input images and their per-pixel Gaussian PSFs visualized on sampled coordinates, respectively. Between the columns, we show deblurring results of different methods. Green and red boxes highlight the regions with large defocus blur and their deblurred results, respectively.
Table 4. Detailed network architecture of the baseline model used for the ablation study.
Table 5. Detailed network architecture of the baseline model embedded with the disparity map estimator used for the ablation study.
| Feature Encoder (F.E.) | type | input | act | k  | c  | s  | p  | N  | output |
|-----------------------|------|-------|-----|----|----|----|----|----|--------|
| conv                  | IB   | lrelu | 3   | 32 | 1  | 1  | 1  | 3  | conv_1 |
| conv                  | IB   | lrelu | 3   | 64 | 2  | 1  | 1  | 2  | conv_2 |
| conv                  | IB   | lrelu | 3   | 64 | 1  | 1  | 2  | 1  | conv_3 |
| conv                  | IB   | lrelu | 3   | 128| 2  | 1  | 1  | 3  | conv_4 |
| conv                  | IB   | lrelu | 3   | 128| 1  | 2  | 1  | 2  | conv_5 |
| IFAN                 | eF   | -     | -   | -  | -  | -  | -  | -  | eF     |
| conv                  | eF   | lrelu | 3   | 128| 1  | 1  | 1  | 1  | conv   |
| RES*                  | conv | -     | -   | -  | -  | -  | -  | -  | RES    |
| conv                  | res  | lrelu | 3   | 129| 1  | 1  | 1  | 1  | conv   |
| conv                  | res  | res   | 3   | 128| 1  | 1  | 1  | 1  | res    |
| IAC                  | (eB, F_deblur) | lrelu | 3   | 128| 1  | 1  | 1  | 17 | eB_S   |
| conv                  | eB_S | lrelu | 3   | 128| 1  | 1  | 1  | 1  | conv   |
| RES*                  | conv | -     | -   | -  | -  | -  | -  | -  | RES    |
| conv                  | res  | lrelu | 3   | 128| 1  | 1  | 1  | 1  | conv   |
| IAC                  | (eS, F_reblur) | lrelu | 3   | 3  | 1  | 1  | 1  | 17 | eS_S   |
| conv                  | up   | lrelu | 3   | 3  | 1  | 1  | 1  | 1  | conv   |
| sum                   | IB   | -     | -   | -  | -  | -  | -  | -  | IB_S   |
| RBN                  | F_deblur | lrelu | 3   | 32 | 1  | 1  | 1  | 1  | conv   |
| RES*                  | conv | -     | -   | -  | -  | -  | -  | -  | RES    |
| RES*                  | res  | -     | -   | -  | -  | -  | -  | -  | res    |
| sum                   | res  | -     | -   | -  | -  | -  | -  | -  | res    |
| IAC                  | (eS, F_reblur) | lrelu | 3   | 3  | 1  | 1  | 1  | 17 | eS_S   |
| sum                   | I_S  | -     | -   | -  | -  | -  | -  | -  | I_S   |
| identity              | input | -     | -   | -  | -  | -  | -  | -  | -      |
| conv                  | conv' | lrelu | 3   | c  | 1  | 1  | 1  | 1  | conv   |
| conv                  | conv' | lrelu | 3   | c  | 1  | 1  | N  | conv | conv   |
| sum                   | conv' | lrelu | -   | -  | -  | -  | -  | -  | conv'  |
| sum                   | input | -     | -   | -  | -  | -  | -  | -  | res    |
| dconv                 | input[0] | lrelu | 4   | c  | 2  | 1  | 1  | 1  | dconv  |
| sum                   | input[1] | -     | -   | -  | -  | -  | -  | -  | sum    |
| RES*                  | sum   | -     | -   | c  | -  | -  | -  | 1  | res    |
| RES*                  | res   | -     | -   | c  | -  | -  | 1  | 1  | up     |

Table 6. Detailed network architecture of the model with the filter predictor and IAC used for the ablation study.
For both training and testing

| Feature Encoder (F.E.) | type | input | act | k | c | s | p | N | output |
|-----------------------|------|-------|-----|---|---|---|---|---|--------|
| conv                  | $I_B$| lrelu | 3   | 32| 1 | 1 | 3 | 1 | $I_B$  |
| conv                  | $I_B$| lrelu | 3   | 64| 2 | 1 | 1 | 2 | $I_B$  |
| conv                  | $I_B$| lrelu | 3   | 64| 1 | 1 | 2 | 1 | $I_B$  |
| conv                  | $I_B$| lrelu | 3   | 128| 2 | 1 | 1 | 1 | $I_B$  |
| conv                  | $I_B$| lrelu | 3   | 128| 1 | 1 | 2 | 1 | $I_B$  |
| conv                  | $I_B$| lrelu | 3   | 128| 2 | 1 | 1 | 2 | $I_B$  |
| conv                  | $I_B$| lrelu | 3   | 128| 1 | 1 | 2 | 1 | $I_B$  |

| F.E. | $I_B$ | - | - | - | - | - | - | $e_F$ |
|------|-------|---|---|---|---|---|---|-------|
| conv | $e_F$ | lrelu | 3 | 128 | 1 | 1 | 1 | conv |
| RES* | conv | - | - | 128 | - | - | 2 | res |
| RES* | res  | - | - | 128 | - | - | 2 | res |
| conv | res  | lrelu | 3 | 128 | 1 | 1 | 1 | conv |
| conv | lrelu | 3 | 128 | 1 | 1 | 1 | conv |
| RES* | conv | - | - | 128 | - | - | 2 | res |
| RES* | res  | - | - | 128 | - | - | 2 | res |
| conv | res  | lrelu | 3 | 128 | 1 | 1 | 1 | conv |
| RES* | conv | - | - | 128 | - | - | 2 | res |
| RES* | res  | - | - | 128 | - | - | 1 | conv |
| conv | res  | lrelu | 3 | 128 | 1 | 1 | 1 | conv |
| RES* | conv | - | - | 128 | - | - | 2 | res |
| RES* | res  | - | - | 128 | - | - | 2 | res |
| conv | res  | lrelu | 3 | 128 | 1 | 1 | 1 | conv |

| IAC | $(e_B, F_{deblur})$ | lrelu | 3 | 128 | 1 | 1 | 17 | $e_B$ |
|-----|---------------------|-------|---|-----|---|---|----|------|
| conv | $e_B$ | lrelu | 3 | 128 | 1 | 1 | 1 | conv |
| RES* | conv | - | - | 128 | - | - | 3 | res |
| conv | lrelu | 3 | 128 | 1 | 1 | 1 | conv |
| UP*  | (res, conv) | - | - | 128 | - | - | - | up |
| UP*  | (up, conv) | - | - | 64 | - | - | - | up |
| conv | up | lrelu | 3 | 1 | 1 | 1 | 17 | $I_B$ |
| sum  | $I_B$ | - | - | - | - | - | - | $I_B$ |

| RBN | $(I_S, F_{reblur})$ | lrelu | 3 | 3 | 1 | 1 | 17 | $I_S$ |
|-----|--------------------|-------|---|---|---|---|----|------|
| conv | $F_{deblur}$ | lrelu | 3 | 32 | 1 | 1 | 1 | conv |
| RES* | conv | - | - | 32 | - | - | 2 | res |
| RES* | res  | - | - | 32 | - | - | 2 | res |
| conv | lrelu | 3 | 357 | 1 | 1 | 1 | conv |
| sum  | $I_S$ | - | - | - | - | - | - | $I_S$ |

| identity | input | - | - | - | - | - | - | conv' |
|----------|-------|---|---|---|---|---|---|-------|
| conv      | $I_B$ | lrelu | 3 | c | 1 | 1 | 1 | conv' |
| conv      | $I_B$ | lrelu | 3 | c | 1 | 1 | N | conv' |
| sum       | $I_B$ | lrelu | - | - | - | - | - | conv' |
| sum       | input | - | - | - | - | - | - | res   |
| dconv     | input[0] | lrelu | 4 | c | 2 | 1 | 1 | dconv |
| sum       | input[1] | - | - | - | - | - | - | sum   |
| RES*      | sum  | - | - | c | - | - | 1 | res   |
| RES*      | res  | - | - | c | - | - | 1 | up    |

Table 7. Detailed network architecture of our final model.
Figure 15. **Network architectures used for the ablation study (Table 1 and Fig. 5 in the main paper).** From top to bottom: architectures for (a) the baseline model, (b) baseline model embedded with the disparity map estimator, (c) proposed IFAN without the disparity map estimator, and (d) proposed IFAN.
References

[1] Abdullah Abuolaim and Michael Brown. Defocus deblurring using dual-pixel data. In ECCV, 2020.

[2] Georgios Evangelidis and Emmanouil Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. IEEE TPAMI, 30(10):1858–1865, 2008.

[3] Junyong Lee, Sungkill Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In CVPR, 2019.

[4] Jaesung Rim, Lee Haeyun, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking de-blurring algorithms. In ECCV, 2020.

[5] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In CVPR, 2014.