Defocus Blur Detection via Multi-Stream Bottom-Top-Bottom Fully Convolutional Network

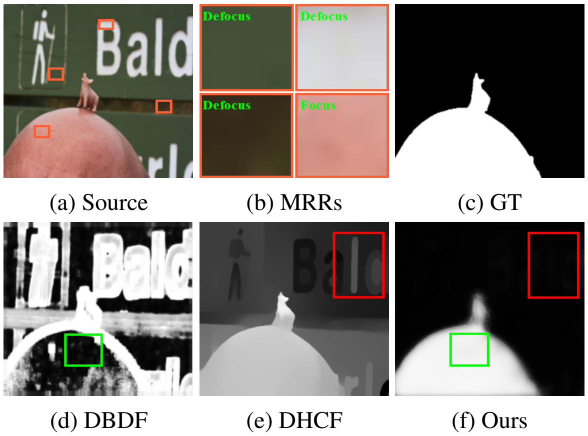
Wenda Zhao[1], Fan Zhao[2], Dong Wang[1]*, Huchuan Lu[1]
[1] Dalian University of Technology, China
[2] Dalian Institute of Chemical Physics, Chinese Academy of Sciences, China
{zhaowenda, wdice, lhchuan}@dlut.edu.cn, zhaofan@dicp.ac.cn

Abstract

Defocus blur detection (DBD) is the separation of in-focus and out-of-focus regions in an image. This process has been paid considerable attention because of its remarkable potential applications. Accurate differentiation of homogeneous regions and detection of low-contrast focal regions, as well as suppression of background clutter, are challenges associated with DBD. To address these issues, we propose a multi-stream bottom-top-bottom fully convolutional network (BTBNet), which is the first attempt to develop an end-to-end deep network for DBD. First, we develop a fully convolutional BTBNet to integrate low-level cues and high-level semantic information. Then, considering that the degree of defocus blur is sensitive to scales, we propose multi-stream BTBNets that handle input images with different scales to improve the performance of DBD. Finally, we design a fusion and recurrent reconstruction network to recurrently refine the preceding blur detection maps. To promote further study and evaluation of the DBD models, we construct a new database of 500 challenging images and their pixel-wise defocus blur annotations. Experimental results on the existing and our new datasets demonstrate that the proposed method achieves significantly better performance than other state-of-the-art algorithms.

1. Introduction

Defocus blur is caused by a wide aperture that prevents light rays converging when the scene point is not at the camera's focus distance. This effect is extremely common in images captured using optical imaging systems. Defocus blur detection (DBD) is performed to separate blurred and non-blurred regions for a given image. Accurate detection of blurred or non-blurred regions is important and practical in many applications, including salient object detection [10], blur region segmentation [12], defocus magnification [2], image restoration [34], to name a few.



(a) Source (b) MRRs (c) GT

(d) DBDF (e) DHCF (f) Ours

Figure 1. A challenging example for defocus blur detection (DBD). (a)-(f): source image, magnified rectangular regions (MRRs), ground truth (GT), DBDF [20], DHCF [17], and our DBD map.

Based on the adopted image features, the DBD methods can be generally divided into two categories: methods based on hand-crafted features; and methods based on deep learning features. The former one exploits low-level defocus blur cues, such as gradient and frequency. Gradient-based methods [18, 16, 36, 38, 23, 15] are motivated by the fact that blur directly suppresses image gradients. Thus, the gradient distribution in a clear region tends to include more heavy-tail components. From the perspective of frequency [17, 27, 20, 35, 28, 37], blur attenuates high frequency components. These methods are often effective in simple DBD scenarios, however, they cannot well handle the following challenging cases.

First, low-level defocus blur cues are difficult to use in differentiating homogeneous regions where the in-focus smooth regions and blurred smooth regions do not contain structural information (e.g., edges and textures). Examples are shown in the orange rectangular regions of Figure 1 (a) and the enlarged regions are illustrated in Figure 1 (b).

*Corresponding Author

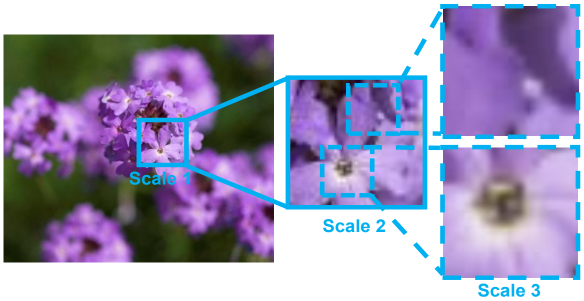


Figure 2. An example of multi-scale blur perception. This figure illustrates that the blur confidence is highly related to scales. A clear image patch (e.g., scale 1) can be regarded as blurry depending on the size of the patch (e.g., scale 3), and vice versa.

Methods based on hand-crafted features cannot effectively capture semantic contexts hidden in an in-focus smooth region, causing erroneous detection in the homogeneous focal region (shown in the green bounding box in Figure 1 (d)).

Second, these cues can hardly detect low-contrast focal regions and suppress the background clutter. This limitation is due to the local measurement of hand-crafted features without considering global semantic information.

Deep convolutional neural networks (CNNs) have successfully overcome the limitations of traditional hand-crafted features in various computer vision tasks, such as object detection [11], image classification [29], image denoising [32], image super-resolution [6], saliency detection [13, 33] and object tracking [14, 24]. In [17], Park *et al.* propose a CNN-based patch-level method to solve the DBD problem. However, this method cannot accurately distinguish low-contrast focal regions and suppress the background clutter (shown in the red bounding box in Figure 1 (e)) since multiple stages of spatial pooling and convolutional layers progressively downsample the initial image patch, thereby resulting in losing much of the fine image structure. In this work, we develop a novel pixel-level multi-stream bottom-top-bottom fully convolutional network for DBD. Our pixel-level method needs to be run once on the input image to produce a complete DBD map with the same pixel resolution as the input image, providing a basic condition to achieve accurate DBD.

Accurate separation of in-focus and out-of-focus image regions requires the simultaneous extraction and combination of high-level semantic information with low-level image details. High-level semantic information should be transformed to bottom layers to assist in the better location of the defocus region. Meanwhile, bottom layers can learn rich low-level features that help refine the sparse and irregular detection maps from top layers. Therefore, we propose a bottom-top-bottom network (BTBNet) by facilitating the flow back of the network information stream. BTBNet effectively merges high-level semantic knowledge encoded in the bottom-top stream with the spatially rich information of low-level features encoded in the top-bottom stream step by step. Based on the observation that the image scale greatly influences the clarity of an image (Figure 2), we use a multi-stream BTBNet to obtain blur detection maps from different scales. Then, we develop a fusion and recursive reconstruction network (FRRNet) to recursively refine the preceding blur detection maps. The proposed method can accurately distinguish homogeneous regions and suppress the background clutter (Figure 1 (f)).

The major contributions of this paper can be summarized as follows:

- A new multi-stream bottom-top-bottom fully convolutional network is proposed to infer a pixel-level defocus blur detection map directly from the raw input image. To our knowledge, this work is the first attempt to exploit an end-to-end deep network in dealing with the DBD problem. The proposed model integrates multi-scale and multi-level features to accurately differentiate homogeneous regions and detect low-contrast focal regions as well as suppress the background clutter.
- A BTBNet is proposed to effectively merge high-level semantic knowledge encoded in the bottom-top stream with low-level features encoded in the top-bottom stream step by step. Moreover, a FRRNet is designed to gradually refine the multi-scale blur detection maps and successfully output a final DBD map.
- A new challenging dataset is constructed for further study and evaluation of DBD. The proposed method has been successfully validated using both new and existing datasets. We will make our dataset publicly available, which promotes further research and evaluation.

2. Related Work

DBD has been widely investigated in computer vision. Previous related works on DBD can be roughly summarized into two categories based on adopted features, namely, hand-crafted features and deep learning features for DBD.

2.1. Hand-crafted Features for DBD

DBD methods with hand-crafted features usually utilize gradient and frequency features. More strong gradients are usually found in a clear image patch than in a blurry one. The ratio of strong gradient components in an image patch can be used to measure the sharpness of the image [7, 25]. For instance, Pang *et al.* [16] develop a kernel-specific feature for blur detection, which involves the multiplication of the variance of filtered kernels and that of filtered patch gradients. Su *et al.* [23] design a blur metric based on singular value distributions and combine the gradient distribution pattern of the alpha channel to detect the blurred region.

Frequency-based methods exploit the fact that an in-focus image has more high-frequency components than an out-of-focus image, thus, the ratio of high-frequency components in an image patch can measure the sharpness of this image. Golestaneh *et al.* [8] propose a spatially-varying blur detection method based on a high-frequency multi-scale fusion and sort transform of gradient magnitudes to determine the level of blur at each location. Tang *et al.* [27] present a blur metric based on the log averaged spectrum residual to obtain a coarse blur map. Shi *et al.* [20] utilize the Fourier domain descriptors with other local distributions and filters as image features to conduct blur detection.

Several other hand-crafted feature-based methods are also available [31, 21, 4, 26]. Yi *et al.* [31] propose a sharpness metric based on the distribution of uniform local binary patterns in blurred and non-blurred image regions for defocus blur segmentation. Shi *et al.* [21] exploit sparse representation and image decomposition to directly establish correspondence between sparse edge representation and blur strength estimation. Couzinie-Devy *et al.* [4] present a multilabel energy minimization framework to model the local blur estimators and their smoothness. Although the hand-crafted methods have been demonstrated to be effective in some cases, these methods are not robust enough in complex scenes. Therefore, it is necessary to consider high-level image information and context for DBD.

2.2. Deep Learning Features for DBD

Deep CNNs have recently set new standard on a number of visual recognition tasks. However, a limited number of deep learning methods can handle the DBD problem. Park *et al.* [17] combine hand-crafted and deep features to obtain a defocus map of a scene, in which a CNN-based model is used to extract high-dimensional deep features from image patches. Then, all features are concatenated to construct a defocus feature vector and fed into a fully connected neural network classifier to determine the degree of defocus. This method measures defocus blur at the patch level where C-NNs are run thousands of times to obtain the blur score of every patch. Thus this process is very time consuming.

In this work, we utilize a fully convolutional network for dense DBD. Low-level cues and high-level semantic information are integrated at the pixel level. In addition, we consider the important influence of scales on defocus blur (Figure 2) to actualize the multi-stream BTBNet for obtaining the multi-scale DBD map. Moreover, we design an FRRNet to gradually refine the preceding blur detection maps. The experimental results demonstrate that the proposed method performs better than the other state-of-the-art approaches.

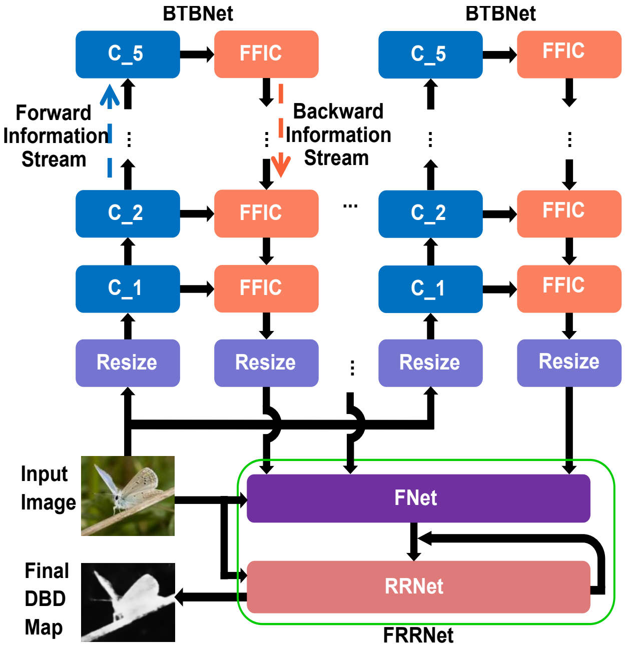


Figure 3. The pipeline of our DBD algorithm. Each colorful box is considered as a feature block. The arrows between blocks indicate the information stream. Given an input image, its multi-scale versions generated by the resize operation are first encoded in the bottom-top stream by a modified VGG16 model [22], respectively. Then, the integration of bottom-top and top-bottom streams is performed by feedback and forward information combination modules (FFICs). After that, the resize operation is employed to ensure generated DBD maps with the same resolution as the original input image. Finally, FRRNet consisting of the fusion network (FNet) and recursion reconstruction network (RRNet) is used to refine the predicted DBD maps, generating the final DBD map.

3. Multi-Stream Bottom-Top-Bottom Fully Convolutional Network

In this work, we exploit an end-to-end fully convolutional network to extract and integrate multi-level multi-scale features for DBD. The entire architecture of our method is illustrated in Figure 3. The construction of the bottom-top-bottom fully convolutional network (BTBNet) is presented in Section 3.1. Then, the fusion and recurrent reconstruction network (FFRNet) is described in Section 3.2. The process of model training is introduced in Section 3.3.

3.1. BTBNet

We aim to design an end-to-end BTBNet that can be viewed as a regression network mapping an input image to a pixel-level blur detection map. Several concepts have been considered to conceive such architecture. First, the network should be deep enough to produce a big receptive field to detect defocus blur at different levels. Second, the network requires low-level cues as well as high-level semantic in-

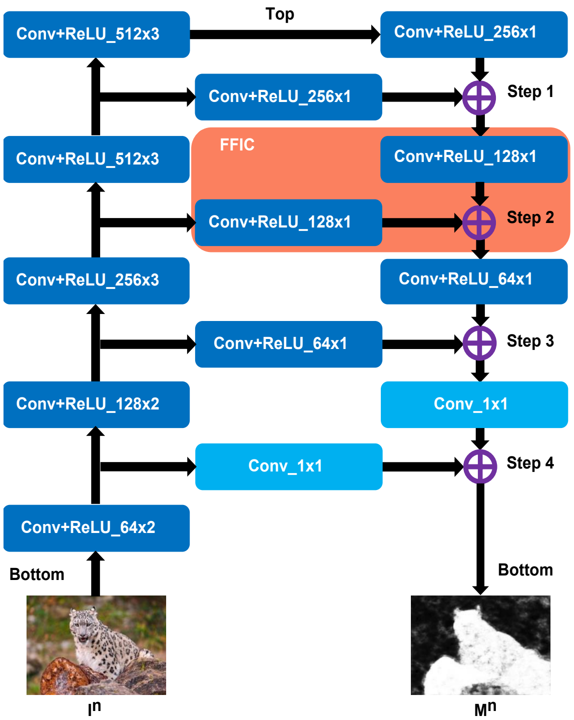


Figure 4. The architecture of BTBNet. Information of an input image $\mathbf{I}^n$ passes from the bottom layer to the top layer with a series of feature blocks, thereby resulting in high-level semantic information. Then, high-level semantic information passes from the top layers down and is integrated with high-resolution low-level cues by the FFIC module, ultimately producing the DBD map $\mathbf{M}^n$. $\oplus$ stands for the element-wise addition.

formation to improve the accuracy of DBD. Finally, fine-tuning an existing deep model is greatly preferred, as the labeled defocus blur images are not sufficient to train such a deep network from scratch.

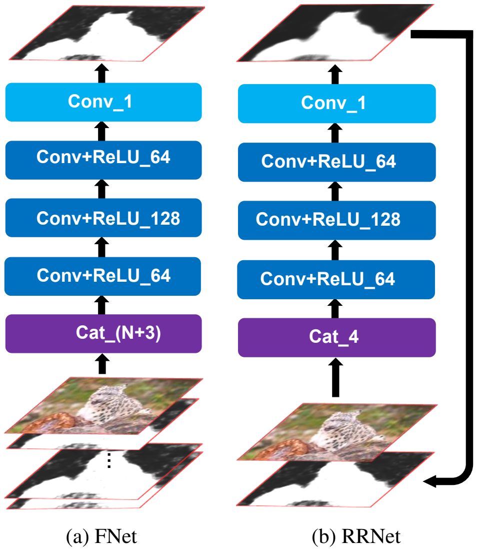
Basic BTBNet. We choose the VGG16 model [22] as our pre-trained network and modify it to meet our requirements. VGG16 has delivered remarkable performance in many recognition tasks. However, it has obvious limitations in dealing with DBD. Multiple stages of spatial pooling progressively downsample the initial image, resulting in considerable loss of the fine image structure. This process is beneficial for the classification task which does not need spatial information but presents challenges for densely separating in-focus and out-of-focus image regions. To transform the original VGG16 model into a fully convolutional network, which serves as our bottom-top backbone network, we delete the top three fully connected layers of VGG16. We also delete the five pooling layers to increase the density of the bottom-top DBD map. Thus, the output resolution of the transformed VGG network is the same with the original input resolution, as shown in Figure 4.

To augment the backbone network with a top-bottom refinement stream, we design a step-wise feedback process (Figure 4). Between each block of the bottom-top backbone network, the feedback information is combined with forward information step by step. The integration of feedback and forward information is achieved by element-wise addition. Before the information integration in each step, we attach a extra convolutional (Conv) with rectified linear unite (ReLU) layer on both bottom-top and top-bottom streams. The extra layers have 3×3 kernels and 256, 128, 64, and 1 channels, respectively. The final output is a DBD map with the same resolution as the original input image.

Multi-Stream BTBNet. Considering that defocus blur is sensitive to scales, we replicate the designed BTBNet repeatedly, with one replicate for one scale. Specifically, an input image is resized to multiple different scales. Each scale $\mathbf{I}^n$ $(n = 1, 2, ..., N)$ of the input image passes through one of these replicated BTBNets, and a DBD map $\mathbf{M}^n$ in the same resolution of scale $\mathbf{I}^n$ is produced. Then, these DBD maps are resized to the same resolution as the raw input image using bilinear interpolation (see Figure 3).

3.2. FRRNet

Our FRRNet consists of two sub-networks, namely, fusion network (FNet) and recurrent reconstruction network (RRNet). FNet merges the DBD maps generated by the multi-stream BTBNet, yielding a DBD map $\mathbf{M}^f$ with improved spatial coherence. Then, RRNet gradually recursively refines the DBD map $\mathbf{M}^f$ to obtain a final DBD map $\mathbf{M}_{final}$. The detailed network structures of FNet and RRNet are shown in Figure 5 (a) and Figure 5 (b), respectively.



(a) FNet

(b) RRNet

Figure 5. The architecture of FNet and RRNet models.

FNet. For fusing multi-stream probability maps, Chen *et al.* [3] propose an effective soft weight-based model, where the final output $\widehat{\mathbf{M}^f}$ is produced by the weighted sum of probability maps across all the scales,

$$\widehat{\mathbf{M}^f} = \sum_{n=1}^{N} w_n \cdot \mathbf{M}^n, \tag{1}$$

where w_n denotes the weight of the DBD result $\mathbf{M}^n$ generated from the n-stream BTBNet, and N is the total number of DBD maps. This model only assigns linear weights to the multi-stream DBD maps. Thus, it cannot effectively reconstruct the incomplete foreground information and suppress the background clutter, as shown in Figure 6 (c).

In this work, we propose a simple but effective FNet. As shown in Figure 5 (a), in addition to multi-stream DBD maps, we exploit dense spatial information of source image to improve spatial coherence of the fused map. The multi-stream DBD maps ($\mathbf{M}^1$, $\mathbf{M}^2$,..., $\mathbf{M}^N$) and the source image $\mathbf{I}^1$ are first concatenated into a single (N+3)-channel feature map $\mathbf{F}^0$. Then, this map is fed to a series of Conv and ReLU layers. The Conv layers have 3×3 kernels and 64, 128, 64, and 1 channels, respectively. The final output after Conv is a fused DBD map $\widehat{\mathbf{M}^f}$ with the same resolution as the source image. The formula of our FNet can be written as follows:

$$\mathbf{F}^0 = cat(\mathbf{M}^1, \mathbf{M}^2, ..., \mathbf{M}^N, \mathbf{I}^1), \tag{2}$$

$$\mathbf{F}^t = max(0, W_t * \mathbf{F}^{t-1} + b_t), \tag{3}$$

$$\widehat{\mathbf{M}^f} = W_T * \mathbf{F}^{T-1} + b_T, \tag{4}$$

where W_t and b_t denote the convolution filter and bias of t-th ($t = 1, 2, ..., T$) Conv layer, respectively. $\mathbf{F}^t$ is the multi-channel feature generated from the t-th Conv layer. The FNet nonlinearly integrates the multi-stream DBD maps and exploits the dense spatial information of the source image. Figure 6 (d) demonstrates that the proposed FNet can generate smoother results with the pixel-wise accuracy.

RRNet. Although FNet improves the spatial coherence of the fused DBD map, noise inevitably occurs when the input image has low-contrast foreground or cluttered background, shown in Figure 7 (c). Thus, we introduce a recurrent reconstruction network (RRNet) for the subsequent refinement. Figure 5 (b) illustrates the network structure of RRNet, which has the same architecture as FNet but with different parameters. In each iteration, we feed forward both the source image and the input DBD map through the RRNet to obtain the refined DBD map, which in turn serves as the input DBD map in the next iteration. The input DBD map is initialized as the preceding fused DBD map $\widehat{\mathbf{M}^f}$.

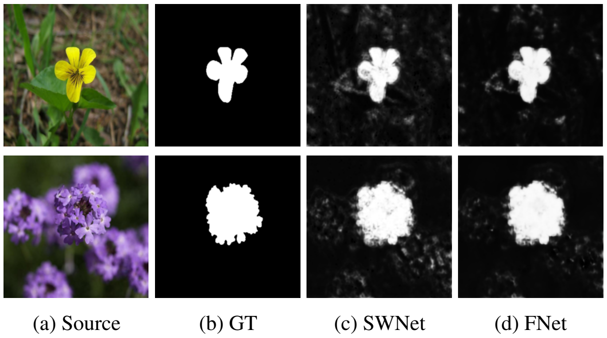


Figure 6. Comparison of multi-stream DBD map fusion results. (a)-(d): source image, ground truth (GT), soft weight-based network (SWNet) [3], and our FNet method.

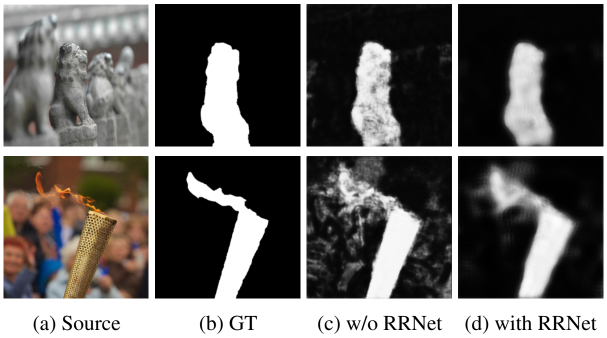


Figure 7. Comparison of DBD results generated from the proposed method without (w/o) and with RRNet.

Let R denote the function modeled by a recursion, the final DBD map $\mathbf{M}_{final}$ can be obtained as follows:

$$\mathbf{M}_{final} = (R \bullet R \bullet \cdots \bullet) R(\widehat{\mathbf{M}^f}; W_r, b_r), \tag{5}$$

where the operator $\bullet$ denotes a function composition. W_r and b_r denote the convolution filter and bias of RRNet. The proposed RRNet can refine the DBD map by correcting its previous mistakes until the final DBD map in the last iteration is produced. In practice, it is enough to use three recurrent steps for achieving satisfactory performance. Figure 7 (d) illustrates that DBD maps generated from our method with RRNet can reconstruct lost information in the foreground and suppress unexpected noise in the background.

3.3. Model Training

The network parameters are determined using training images annotated at the pixel level. The multi-stream BTBNet and FRRNet are jointly trained, and their parameters are optimized using standard stochastic gradient descent. We first initialize the bottom-top backbone network with a VGG16 network pretrained on ImageNet [5] and the top-bottom stream and FRRNet with random values. After that, we jointly fine-tune the multi-stream BTBNet and FRRNet.

Given the training set $\{\mathcal{T} = (\mathbf{X}^d, \mathbf{G}^d)\}_{d=1}^{D}$ containing both training image $\mathbf{X}^d$ and its pixel-wise DBD annotation $\mathbf{G}^d$, where D is the number of training patches. The pixel-wise loss function between the network output $\mathbf{M}^d$ and the ground truth $\mathbf{G}^d$ is defined as follows:

$$\begin{aligned} L(\Theta) = -\sum_{\mathcal{T}}\sum_{i,j}\{&\mathbf{G}_{i,j}^{d}\text{log}p(\mathbf{M}_{i,j}^{d} \mid \Theta) \\ &+(1-\mathbf{G}_{i,j}^{d})\text{log}(1-p(\mathbf{M}_{i,j}^{d} \mid \Theta))\}, \end{aligned} \tag{6}$$

where $p(\mathbf{M}_{i,j}^{d} \mid \cdot) = (1+e^{-\mathbf{M}_{i,j}^{d}})^{-1}$, $\mathbf{M}_{i,j}^{d}$ and $\mathbf{G}_{i,j}^{d}$ indicate the d-th network output and ground truth of pixel (i,j), respectively. Θ is the parameter set of all network layers.

To boost the performance of our model, we apply an auxiliary loss at the output of each stream BTBNet. Both main and auxiliary loss functions facilitate optimizing the learning process. Thus, our final loss function combining main and auxiliary losses can be written as follows:

$$L_{final}(\Theta) = L_{main}(\Theta) + \sum_{n=1}^{N} \alpha_n L_{aux}^{n}(\Theta), \tag{7}$$

where N is the number of streams for BTBNet, and α_n is a trade-off parameter that is taken as 1 to balance all losses.

4. Experiments

4.1. Experimental Setup

Datasets. To the best of our knowledge, only one public blurred image dataset [20] (Shi's dataset) is available. This dataset consists of 704 partially defocus blurred images and manually annotated ground truths to study and evaluate DBD models. We divide the Shi's dataset into two parts, that is, 604 for training and the remaining 100 images for testing. Then, we perform data augmentation similar to that in [30]. Specifically, the training set is enlarged to 9664 images by horizontal flipping at each orientation and rotating to 8 different orientations.

In addition, to facilitate the study and evaluation of defocus blur detection (DBD) methods, we construct a new DBD dataset consisting of 500 images with pixel-wise annotations. We note that the proposed dataset is very challenging since numerous images contain homogeneous regions, low-contrast focal regions and background clutter. We will make our dataset publicly available for further DBD researches.

Implementation. Our network has been implemented on the basis of Caffe [9]. The training process is conducted by optimizing the classification objective using the mini-batch gradient descent method with a batch size of 1.

First, we initialize the bottom-top backbone network with the VGG16 model trained on ImageNet [22], and the top-bottom stream and FRRNet with random values.

Second, we fine-tune our model by pre-training it on a simulated image dataset. Specifically, we collect 2000 clear images from the Berkeley segmentation dataset [1] and uncompressed colour image dataset [19]. We adopt a Gaussian filter for each image to smooth half of the image as the out-of-focus blur region, and the remaining half as the in-focus region. Then, four blurred versions can be obtained by smoothing regions of different positions (up, down, left and right) for each image. For each blurred version, we use a Gaussian filter with a standard deviation of 2 and a window of 7×7 to repeatedly blur the image five times. Therefore, for each image, we can obtain 20 simulated images (four blurred versions and five different blurring levels for each version). Thus, with the data augmentation above,the adopted pre-trained image dataset contains 640K images.

Finally, we fine-tune our model on the Shi's training dataset. We set the initial learning rate to 0.0001 for the backbone network and 0.001 for newly added layers. The momentum parameter is 0.9 and the weight decay is 0.0005.

The proposed model is trained on a workstation with an Intel 3.4GHz CPU with 32G memory and a GTX1080Ti GPU with 11G memory, and training is completed after approximately 5 days. Approximately 25s is needed to generate the DBD map for a testing image with 320×320 pixels.

4.2. Evaluation criteria

First, we use the precision-recall (PR) curve [8, 17, 20] to evaluate different methods. All DBD maps are binarized at every integer threshold in the range of [0, 255]. Compared with the binary ground-truth masks, pairs of precision and recall values are computed to report all PR curves.

Second, we compute the average precision, recall and F-measure values, where every DBD map is binarized with an adaptive threshold. The threshold is determined to be 1.5 times the mean value of the DBD map. The F-measure is an overall performance measurement calculated as follow:

$$F = \frac{(1+\zeta^2) \cdot Precision \cdot Recall}{\zeta^2 \cdot Precision + Recall}, \tag{8}$$

where ζ^2 is 0.3. A larger F value means a better result.

Third, we report the mean absolute error (MAE) which is calculated as the average pixel-wise absolute difference between the binary ground truth $\mathbf{G}$ and the DBD map $\mathbf{M}_{final}$,

$$MAE = \frac{1}{W \times H}\sum_{x=1}^{W}\sum_{y=1}^{H}\left|\mathbf{G}(x,y) - \mathbf{M}_{final}(x,y)\right|, \tag{9}$$

where x, y stand for pixel coordinates. W and H denote the width and height of the DBD map $\mathbf{M}_{final}$, respectively. A smaller MAE value usually means a more accurate result.

4.3. Comparison with state-of-the-art methods

We compare our method with six state-of-the-art methods, including discriminative blur detection features (DBD-

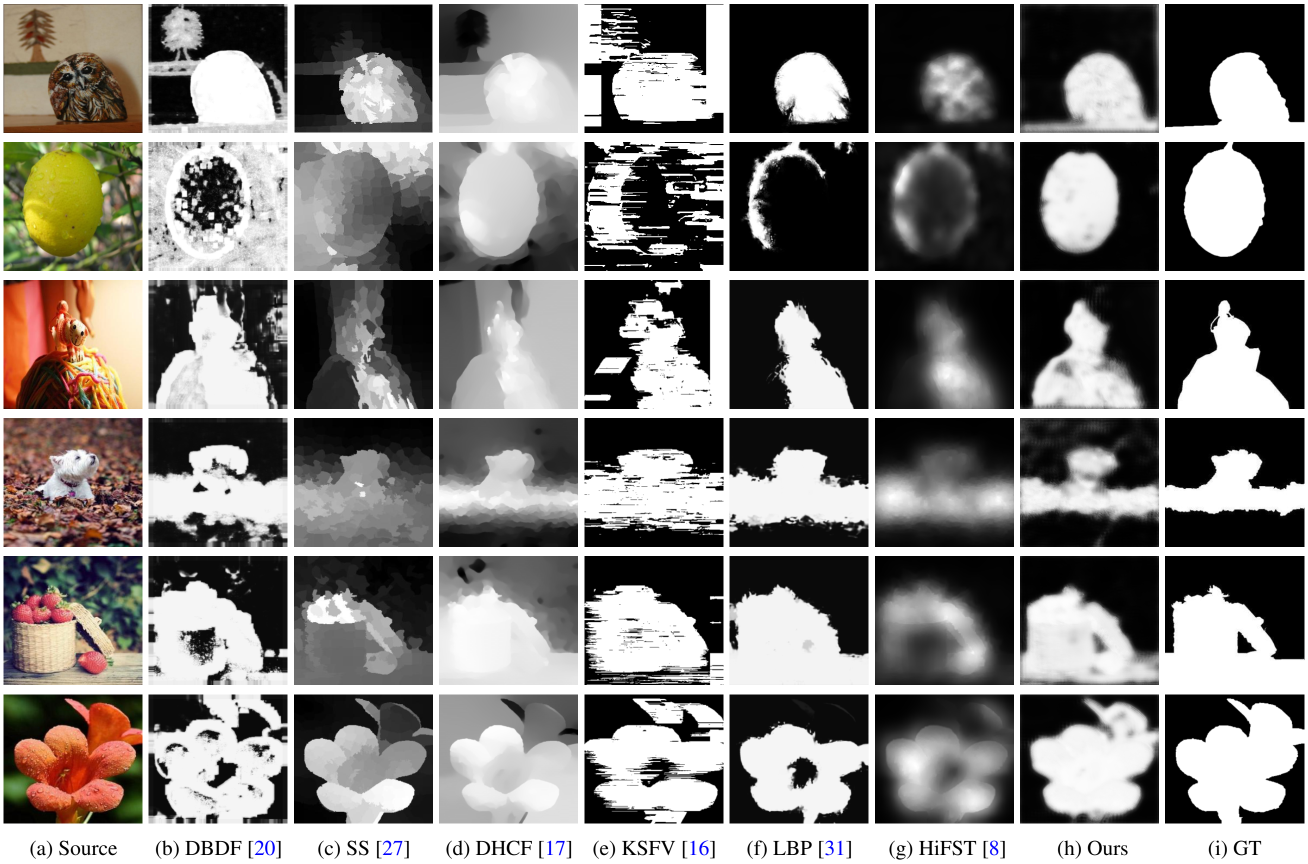
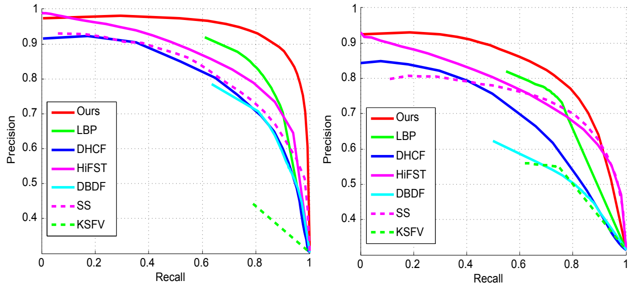

(a) Source (b) DBDF [20] (c) SS [27] (d) DHCF [17] (e) KSFV [16] (f) LBP [31] (g) HiFST [8] (h) Ours (i) GT

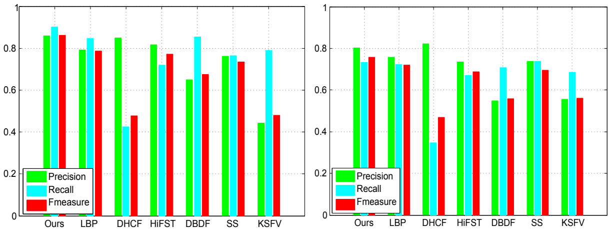
Figure 8. Visual comparison of DBD maps generated from the proposed method and other state-of-the-art ones. The ground truth is shown in the last column. The first four sources are selected from the Shi's dataset. The last four sources are chosen from our dataset. It can be seen from that our method consistently produces DBD maps closest to the ground truth.



(a) Shi's dataset (b) Our dataset

Figure 9. Comparison of precision-recall curves of seven state-of-the-art methods using (a) Shi's dataset and (b) our dataset. The proposed method outperforms other methods on both datasets.

F) [20], spectral and spatial approach (SS) [27], deep and hand-crafted features (DHCF) [17], kernel-specific feature vector (KSFV) [16], local binary patterns (LBP) [31] and high-frequency multi-scale fusion and sort transform of gradient magnitudes (HiFST) [8]. We use the original implementation of these methods with recommended parameters.



(a) Shi's dataset (b) Our dataset

Figure 10. Comparison of precision, recall and F-measure using (a) Shi's dataset and (b) our dataset. The proposed method achieves the highest F-measure on both datasets.

Qualitative Evaluation. A visual comparison is provided in Figure 8. The proposed method performs well in various challenging cases (e.g., homogeneous regions, low-contrast in-focus regions, and cluttered background), yielding DBD maps closest to the ground truth maps.

Quantitative Evaluation. PR curves and F-measure values are reported in Figures 9 and 10, from which we can

Table 1. Quantitative comparison of F-measure and MAE scores. The best two results are shown in **red** and **blue** colors, respectively.

Dataset	Metric	DBDF [20]	SS [27]	DHCF [17]	KSFV [16]	LBP [31]	HiFST [8]	Ours
Shi's dataset	F-measure	0.675	0.734	0.477	0.480	**0.787**	0.772	**0.867**
	MAE	0.290	0.229	0.372	0.357	**0.136**	0.219	**0.107**
Our dataset	F-measure	0.558	0.695	0.468	0.560	**0.719**	0.687	**0.761**
	MAE	0.381	0.291	0.410	0.275	**0.193**	0.248	**0.194**

Table 2. Ablation analysis using F-measure and MAE values. The best two results are shown in **red** and **blue** colors, respectively. 'Ours' stands for the final model, i.e., BTBNet(3S)+FNet+RRNet.

Method	Shi's dataset		Our dataset	
	F-measure	MAE	F-measure	MAE
VGGNet(FC)	0.797	0.222	0.643	0.273
BTBNet(1S)	0.853	0.144	0.730	0.237
BTBNet(2S)+FNet	0.853	0.130	0.731	0.233
BTBNet(3S)+FNet	**0.854**	**0.130**	**0.732**	**0.230**
BTBNet(4S)+FNet	0.853	0.135	0.731	0.231
Ours	**0.867**	**0.107**	**0.761**	**0.194**

see that the proposed method achieves the top performance over both datasets and all evaluation metrics. In addition, we compare our method with other competing ones in terms of F-measure and MAE values, and the results are shown in Table 1. Especially for the F-measure metric, our method improves the second best one (LBP [31]) by 10.2% and 5.8% over the Shi's dataset and our dataset, respectively.

4.4. Ablation Studies

Effectiveness of BTBNet. To demonstrate the superiority of BTBNet, we train a VGG16-based fully convolutional network denoted as VGGNet(FC) for comparison. To be specific, we remove the top three fully connected layers of VGG16 and five pooling layers to make the output resolution be same as the input resolution. VGGNet(FC) is trained using the same setting as BTBNet. We compare the implemented VGGNet(FC) with one-stream BTBNet, named as BTBNet(1S). The quantitative results are reported in the first two rows in Table 2. Using F-measure values, our BTBNet(1S) improves the VGGNet(FC) method by 7.0% and 13.5% over the Shi's dataset and our dataset, respectively. Moreover, our BTBNet(1S) lowers the MAE scores significantly on both datasets.

Effectiveness of multi-stream BTBNet. As described in Section 3.1, a multi-stream BTBNet is utilized to detect defocus blur from multi-scale input images. Here, we train four models for comparison to analyze the relative contributions of the different BTBNet streams. These models are as follows: one-stream BTBNet with input image scale $s_1 = \{1\}$, BTBNet(1S); two-stream BTBNet with input image scale $s_2 = \{1, 0.8\}$, BTBNet(2S); three-stream BTBNet with input image scale $s_3 = \{1, 0.8, 0.6\}$, BTBNet(3S); and four-stream BTBNet with input image scale $s_4 = \{1, 0.8, 0.6, 0.4\}$. A detailed comparison of their performance using F-measure and MAE values are reported in Table 2. It can be seen from this table that the multi-stream mechanism effectively improves the detection performance and three-stream BTBNet achieves the best performance.

Effectiveness of RRNet. By comparing the last two rows in Table 2, we can see that our model with RRNet performs much better than that without RRNet on both datasets. The underlying reason is that RRNet is able to further suppress foreground noise and background clutter (see Section 3.2).

5. Conclusions

In this paper, we propose a novel end-to-end defocus blur detection (DBD) method based on multi-stream bottom-top-bottom (BTB) fully convolutional network. First, BTBNet effectively combines high-level semantics with low-level image features to produce high-resolution DBD maps. Second, the multi-stream mechanism allows our network to take advantage of different scale information. In addition, RRNet can further recursively refine the DBD map to achieve more accurate results. The extensive experimental results illustrate that the proposed algorithm achieves significantly better performance than other state-of-the-art approaches, especially in the presence of homogeneous regions, low-contrast focal regions and background clutter. In addition, we construct a new dataset including 500 challenging images associated with their pixel-wise blur detection annotations. We believe that this dataset will promote further research and evaluation of DBD models.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant Nos. 61771088, 61725202, 61502070 and 61472060, the China Postdoctoral Science Foundation under Grant No. 2017M611221, and the Fundamental Research Funds for the Central Universities under Grant Nos. DUT16RC(3)077, DUT2017TB04 and DUT17TD03.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5):898–916, 2011.

[2] S. Bae and F. Durand. Defocus magnification. *Computer Graphics Forum*, 26(3):571–579, 2007.

[3] L. C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2016.

[4] F. Couzinie-Devy, J. Sun, K. Alahari, and J. Ponce. Learning to estimate and remove non-uniform image blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1075–1082, 2013.

[5] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] C. Dong, C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.

[7] J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, 1998.

[8] S. A. Golestaneh and L. J. Karam. Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5800–5809, 2017.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014.

[10] P. Jiang, H. Ling, J. Yu, and J. Peng. Salient region detection by UFO: Uniqueness, focusness and objectness. In *IEEE International Conference on Computer Vision*, pages 1976–1983, 2013.

[11] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 817–825, 2016.

[12] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1699–1712, 2008.

[13] G. Li and Y. Yu. Visual saliency detection based on multiscale deep CNN features. *IEEE Transactions on Image Processing*, 25(11):5012–5024, 2016.

[14] P. Li, D. Wang, L. Wang, and H. Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018.

[15] R. Liu, Z. Li, and J. Jia. Image partial blur detection and classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[16] Y. Pang, H. Zhu, X. Li, and X. Li. Classifying discriminative features for blur detection. *IEEE Transactions on Cybernetics*, 46(10):2220–2227, 2016.

[17] J. Park, Y. W. Tai, D. Cho, and I. S. Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1736–1745, 2017.

[18] E. Saad and K. Hirakawa. Defocus blur-invariant scale-space feature extractions. *IEEE Transactions on Image Processing*, 25(7):3141–3156, 2016.

[19] G. Schaefer and M. Stich. UCID: An uncompressed color image database. *Storage and Retrieval Methods and Applications for Multimedia*, pages 1–9, 2003.

[20] J. Shi, L. Xu, and J. Jia. Discriminative blur detection features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2972, 2014.

[21] J. Shi, L. Xu, and J. Jia. Just noticeable defocus blur detection and estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–665, 2015.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, pages 1–14, 2014.

[23] B. Su, S. Lu, and C. L. Tan. Blurred image region detection and classification. In *ACM International Conference on Multimedia*, pages 1397–1400, 2011.

[24] C. Sun, D. Wang, H. Lu, and M. Yang. Learning spatial-Aware regressions for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[25] Y. W. Tai and M. S. Brown. Single image defocus map estimation using local contrast prior. In *IEEE International Conference on Image Processing*, pages 1797–1800, 2009.

[26] C. Tang, C. Hou, and Z. Song. Defocus map estimation from a single image via spectrum contrast. *Optics Letters*, 38(10):1706–1708, 2013.

[27] C. Tang, J. Wu, Y. Hou, P. Wang, and W. Li. A spectral and spatial approach of coarse-to-fine blurred image region detection. *IEEE Signal Processing Letters*, 23(11):1652–1656, 2016.

[28] C. T. Vu, T. D. Phan, and D. M. Chandler. S3: A spectral and spatial measure of local perceived sharpness in natural images. *IEEE Transactions on Image Processing*, 21(3):934–945, 2012.

[29] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. HCP: A flexible CNN framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1901–1907, 2016.

[30] S. Xie and Z. Tu. Holistically-nested edge detection. In *IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[31] X. Yi and M. Eramian. LBP-based segmentation of defocus blur. *IEEE Transactions on Image Processing*, 25(4):1626–1638, 2016.

[32] K. Zhang, W. Zuo, Y. J. Chen, D. Y. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[33] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin. Learning uncertain convolutional features for accurate saliency detection. In *IEEE International Conference on Computer Vision*, pages 212–221, 2017.

[34] X. Zhang, R. Wang, X. Jiang, W. Wang, and W. Gao. Spatially variant defocus blur map estimation and deblurring from a single image. *Journal of Visual Communication and Image Representation*, 35:257–264, 2016.

[35] Y. Zhang and K. Hirakawa. Blur processing using double discrete wavelet transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1098, 2013.

[36] J. Zhao, H. Feng, Z. Xu, Q. Li, and X. Tao. Automatic blur region segmentation approach using image matting. *Signal, Image and Video Processing*, 7(6):1173–1181, 2013.

[37] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar. Estimating spatially varying defocus blur from a single image. *IEEE Transactions on Image Processing*, 22(12):4879–4891, 2013.

[38] S. Zhuo and T. Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011.