DeFusionNET: Defocus Blur Detection via Recurrently Fusing and Refining Multi-scale Deep Features

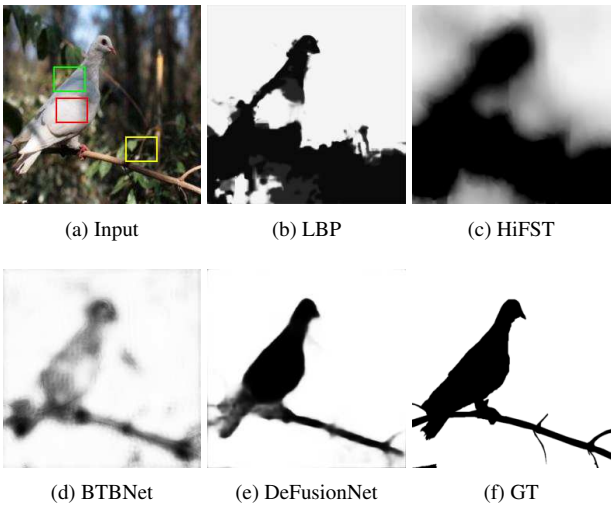
Chang Tang[1], Xinzhong Zhu[2], Xinwang Liu[3], Lizhe Wang[1], Albert Zomaya[4]
[1]School of Computer Science, China University of Geosciences, Wuhan 430074, China
[2]College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China
[3]School of Computer Science, National University of Defense Technology, Changsha 410073, China
[4]School of Information Technologies, University of Sydney, NSW 2006, Australia
{tangchang@cug.edu.cn,zxz@zjnu.edu.cn,xinwangliu@nudt.edu.cn,Lizhe.Wang@gmail.com,albert.zomaya@sydney.edu.au}

Abstract

Defocus blur detection aims to detect out-of-focus regions from an image. Although attracting more and more attention due to its widespread applications, defocus blur detection still confronts several challenges such as the interference of background clutter, sensitivity to scales and missing boundary details of defocus blur regions. To deal with these issues, we propose a deep neural network which recurrently fuses and refines multi-scale deep features (DeFusionNet) for defocus blur detection. We firstly utilize a fully convolutional network to extract multi-scale deep features. The features from bottom layers are able to capture rich low-level features for details preservation, while the features from top layers can characterize the semantic information to locate blur regions. These features from different layers are fused as shallow features and semantic features, respectively. After that, the fused shallow features are propagated to top layers for refining the fine details of detected defocus blur regions, and the fused semantic features are propagated to bottom layers to assist in better locating the defocus regions. The feature fusing and refining are carried out in a recurrent manner. Also, we finally fuse the output of each layer at the last recurrent step to obtain the final defocus blur map by considering the sensitivity to scales of the defocus degree. Experiments on two commonly used defocus blur detection benchmark datasets are conducted to demonstrate the superority of DeFusionNet when compared with other 10 competitors. Code and more results can be found at: http://tangchang.net

1. Introduction

As a common phenomenon, defocus blur occurs when the objects of a scene are not exactly at the camera's focus distance. Defocus blur detection, which aims to detect the out-of-focus regions from an image, has obtained much attention due to its wide potential applications such as image quality assessment [38, 32], salient object detection [9, 34], image deblurring [17, 25], defocus magnification [33, 2] and image refocusing [44, 45], just list a few.



(a) Input (b) LBP (c) HiFST (d) BTBNet (e) DeFusionNet (f) GT

Figure 1. Some challenging cases for defocus blur detection. (a) Input image, defocus blur detection maps obtained by (b) LBP [41], (c) HiFST [1], (d) BTBNet [48], (e) our DeFusionNet, and (f) ground truth (GT).

In the past decades, a variety of defocus blur detection methods have been proposed. Based on the used image features, these methods can be roughly classified into two categories, i.e., traditional hand-crafted features based methods and deep learning based methods. As to the former kind of methods, they often extract features such as gradient and frequency which can model the edge changes since defocus blur usually blunts object edges in an image[15, 50, 29, 37, 46, 49, 25, 19, 35, 24, 21]. Although great success has been achieved by using these tradition-

al hand-crafted features based methods, they still confront several challenges and the detected results are still not very perfect. Firstly, traditional low-level features can not work well for separating the blurred smooth regions which do not contain structural information from the in-focus smooth regions. Secondly, these methods can not well capture the global semantic information which is critical for detecting low-contrast focal regions (as shown in the red rectangular region of Figure 1a) and suppressing the background clutter (as shown in the yellow rectangular region of Figure 1a). In addition, the edge information of in-focus objects have not been well preserved (as shown in the green rectangular region of Figure 1a).

Recently, due to the strong feature extraction and learning capability, deep convolutional neural networks (CNNs) have made remarkable advances in various computer vision tasks, such as image classification [12, 28], object detection [11, 14], object tracking [13, 30, 23], scene semantic segmentation [18, 16, 47], image de-noising [10, 42] and super-resolution [5, 27]. As a result, CNNs are also used for image defocus blur region detection. In [40], a pre-trained deep neural network and a general regression neural network are proposed to classify the blur type and then estimate its parameters. By systematically analyzing the effectiveness of different defocus detection features, Park et al. [21] extracted deep and hand-crafted features in image patches which contain sparse strong edges. However, low-contrast focal regions are still not well distinguished. In addition, a series of spatial pooling and convolution operations result in losing much of the fine details of image structure. In [48], Zhao et al. proposed a multi-stream bottom-top-bottom fully convolutional network (BTBNet), which is the first attempt to develop an end-to-end deep network for defocus blur detection. In BTBNet, low-level cues and high-level semantic information are integrated to promote the final results and a multi-stream strategy is leveraged to handle the defocus degree's sensitivity to image scales. Although significant improvement has been obtained by BTBNet, it uses a forward stream and a backward stream to integrate features from different levels for each image scale, this causes high computational complexity for both network training and testing, and the complementary information of different layers cannot been fully exploited, which causes some background clutters in the final results. In addition, some low-contrast focal areas are still mistakenly detected as defocus blur regions. In this work, we propose a novel efficient pixel-wise fully convolutional network for defocus blur detection via recurrently fusing and refining multi-scale deep features (DeFusionNET). Particularly, we recurrently fuse and refine the deep features across deep and shallow layers in a we summarize the technical contributions of this work as follows:n alternate and cross-layer manner, then the complementary information of features from different layers can be fully exploited for maximized defocus blur detection performance. In detail,

- We design a new efficient pixel-wise fully convolutional network for defocus blur detection from the raw input image. The proposed network fuses and refine multi-scale deep features to effectively suppress the background clutter and distinguish low-contrast focal regions from defocus blur areas.

- Instead of directly refining the detected defocus blur map, we develop a feature fusing and refining module (FFRM) to recurrently refine the features of different layers in an alternate and cross-layer manner. By considering that different layers extract features of different scales for an image, we aggregate the output score maps of different layers at the last recurrent step to generate the final defocus blur map.

- We evaluate our network on two benchmark datasets and compare it with 10 state-of-the-art defocus blur detection methods. The experimental results demonstrate that our method consistently outperforms other competitors on the two datasets. In the meanwhile, our network is very efficient and it takes only less than 0.1s by using a single GTX Titan Xp GPU with 12G memory to generate the defocus blur map for a testing image in the two datasets.

- We aim to set up a benchmark for comparison of various defocus blur detection methods. The results of various methods on different datasets will be publicly released for academic usage.

2. Related Work

2.1. Hand-crafted Features based Methods

Since defocus blur usually degenerates object edges in an image, traditional methods often extract features such as gradient and frequency which can describe the change of edges [6, 31, 50, 33, 4, 32]. Based on the observation that the first few most significant eigen-images of a blurred image patch usually have higher weights (i.e. singular values) than an image patch with no blur, Su et al. [29] detected blur regions by examining singular value information for each image pixels. Shi et al. [25] studied a series of blur feature representations such as gradient, Fourier domain, and data-driven local filters features to enhance discriminative power for differentiating blurred and unblurred image regions. In [19], Pang et al. developed a kernel-specific feature for blur detection, the blur regions and in-focus regions are classified using SVM. Considering that feature descriptors based on local information cannot distinguish the just noticeable

blur reliably from unblurred structures, Shi et al. [26] proposed a simple yet effective blur feature via sparse representation and image decomposition. Yi and Eramian [41] designed a sharpness metric based on local binary patterns and the in- and out-of-focus image regions are separated by using the metric. Tang et al. [36] designed a log averaged spectrum residual metric to obtain a coarse blur map, then an iterative updating mechanism is proposed to refine the blur map from coarse to fine based on the intrinsic relevance of similar neighbor image regions. Golestaneh and Karam [1] proposed to detect defocus blur maps based on a novel high-frequency multiscale fusion and sort transform of gradient magnitudes. Based on the maximum ranks of the corresponding local patches with different orientations in gradient domain, Xu et al. [39] presented a fast yet effective approach to estimate the spatially varying amounts of defocus blur at edge locations, then the complete defocus map is generated by a standard propagation procedure.

Although previous hand-crafted methods have earned great success for defocus blur region detection, they can only work well for images with simple structures but are not robust enough for complex scenes. Therefore, extracting high level and more discriminative features are necessary.

2.2. Deep Learning based Methods

Due to their high level feature extraction and learning power, deep CNNs based methods have refreshed the records of many computer vision tasks [28, 11, 13, 47, 27], including defocus blur detection [21, 48]. In [21], high-dimensional deep features are first extracted by using a CNN-based model, then these features and traditional hand-crafted features are concatenated together and fed into a fully connected neural network classifier for defocus degree determination. Purohit et al. [22] proposed to train two sub-networks which aim to learn global context and local features respectively, then the pixel-level probabilities estimated by two networks are aggregated and feed into a MRF based framework for blur regions segmentation. Zhang et al. [43] proposed a dilated fully convolutional neural network with pyramid pooling and boundary refinement layers to generate blur response maps. Considering that the degree of defocus blur is sensitive to scales, Zhao et al. [48] proposed a multi-stream bottom-top-bottom fully convolutional network (BTBNet) which integrates low-level cues and high-level semantic information for defocus blur detection. Since it uses two streams, i.e., a forward stream and a backward stream, to integrate features from different levels for multiple image scales, the computational complexity for both network training and testing of BTBNet is high. Meanwhile, some low-contrast focal areas still cannot be differentiated.

In this work, we propose an effective and efficient defocus blur detection deep neural network via recurrently fusing and refining multi-scale deep features (DeFusionNET). Instead of directly refining the output score map as many previous deep CNNs based detection methods do, we recurrently refine the features of different layers in DeFusionNET. Particularly, we design a feature fusing and refining module (FFRM) to exploit the complementary information of low-level cues and high-level semantic features by refining them in a cross-level manner, i.e., features from low-level layers are fused and used to refine features extracted from high-level layers, and vice versa. Note that different layers extract features of different scales for an image and the degree of defocus blur is sensitive to image scales, we fuse the output score maps of different layers at the last recurrent step to generate the final defocus blur map. Experimental results demonstrate that the proposed DeFusionNET performs better than other state-of-the-art approaches in terms of both accuracy and efficiency.

3. Proposed DeFusionNET

In this work, we aim to develop an efficient defocus blur detection deep neural network which takes an image as input and output a defocus blur detection map with the same resolution as the input image. Figure 2 shows the entire architecture of our proposed defocus blur detection network.

For an effective defocus blur detection network, it should be power to extract both low-level cues and high-level semantic information for generate the final accurate detected defocus blur map. The low-level features can help refine the sparse and irregular detection regions, while the high-level semantic features can serve to locate the blurry regions as well as suppress background clutters. In addition, there are often some smooth in-focus regions within an object, the high-level semantic information produced by deep layers can avoid these regions being detected as blurry regions. Furthermore, since the defocus degree is sensitive to image scales, the network should be capable of making use of multi-scale features for improving the final results. Finally, the network should be easily to be fine-tuned because there are no sufficient labeled defocus blur images for training such a deep network.

Specifically, we choose the VGG network [28] as our backbone feature extraction network and use the pre-trained VGG16 model to initialize our network. Firstly, we use our network to extract a set of hierarchical features which encode the low-level details and high-level semantic information with different scales of an image. On the one hand, since a series of spatial pooling and convolution operations progressively downsample the resolution of the initial image, the fine details of image structure are inevitably damaged, which is harmful for densely separating in-focus and out-of-focus image regions. On the other hand, the high-level semantic features extracted by deep layers can help to locate the defocus blur regions. Therefore, how to exploit

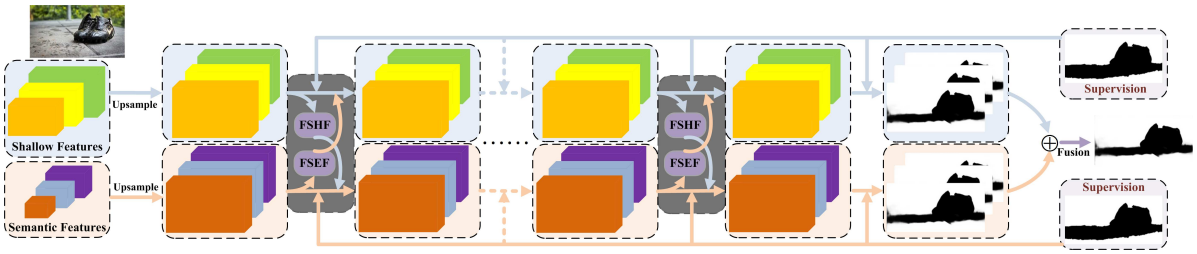


Figure 2. The pipeline of our DeFusionNET. The dark gray block represents the proposed FFRM module. For a given image, we first extract its multi-scale features by using the basic VGG network. Then the features from shallow layers and deep layers are fused as FSHF and FSEF, respectively. Considering the complementary information between FSHF and FSEF, we use them to refine the features of deep and shallow layers in a cross-layer manner. The feature fusion and refinement are performed step by step in a recurrent manner to alternatively refine FSHF, FSEF and the features at each layer (the times of recurrent step is empirically set to 3 in our experiments). In addition, the deep supervision mechanism is imposed at each step and the prediction result of each layer are fused to obtain the final defocus blur map.

the complementary information of features extracted from shallow layers and deep layers to improve the final results is critical. As to the low-level and high-level feature maps, we upsample them to the size of input image by using the deconvolution operation and concatenate them together to form fused shallow features (FSHF) and fused semantic features (FSEF), respectively. In order to refine the detailed information of features at deep layers, we aggregate the FSHF with each deep layer as FSHF encompasses more details of image contents. In order to facilitate the defocus blur region location information of features at shallow layers, we also aggregate the FSEF with each shallow layer as FSEF captures more semantic information of image contents. The feature fusing and aggregating are recurrently carried out in a cross-layer manner. Since different layers extract features with different scales for an image and the degree of defocus blur is sensitive to image scales, the output score maps of different layers at the last recurrent step are fused to generate the final defocus blur map.

3.1. Feature Fusing and Refining Module

The success of deep CNNs owes to its strong capacity of hierarchically extracting abundant semantic as well as fine details information by different layers. For defocus blur region detection, the features represent fine details are necessary since they can benefit to preserve the boundaries between in-focus regions and out-of-focus regions for promoting detection accuracy. The high-level semantic information can serve to accurately locate the defocus blur regions and avoid the smooth in-focus regions being falsely regarded as blur regions, which is also critical. As a result, we can integrate multi-level features to enhance the discrimination ability for defocus blur detection. In deep CNNs, deep layers can capture highly semantic information which describe the attributes of image contents as a whole, while shallow layers focus more on subtly fine details which represent delicate structures of objects, directly fusing the features from different layers for generating final detection results may not be appropriate. In this work, we propose a feature fusing and refining module (FFRM) which integrates high-level semantic features and low-level shallow features separately and refines them in a cross-layer manner. Figure 3 shows the architecture of the proposed FFRM model.

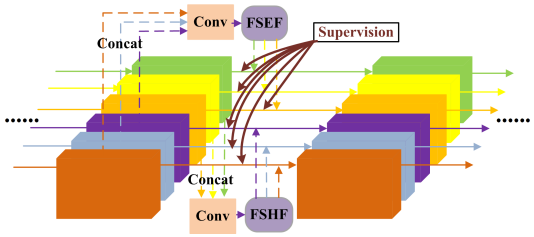


Figure 3. The architecture of the proposed feature fusing and refining module (FFRM).

Supposing there are n total layers in our network, we regard the first m layers as shallow layers and the rest ones as deep layers. For the feature maps generated from each shallow layer, we first upsample them to the size of input image by using the deconvolution operation and concatenate them together, then a convolution layer with 1×1 kernel follows the concatenated feature maps is used to generate FSHF. The FSHF can be mathematically defined as follows:

$$FSHF = ReLU(\mathbf{W}_l * Cat(\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_m)) + \mathbf{b}_l), \quad (1)$$

where $\mathbf{F}_i \in W \times H \times C$ denotes the upsampled feature maps from the i-th layer with C channels; $W \times H$ is the resolution of input image; *Cat* represents the concatenation operation across channels; $*$ represents convolution operation; $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weights and bias of the convolution need to be learned during training and *ReLU* is the ReLU activation function [12].

Similarly, the high-level semantic features are fused to form FSEF as follows:

$$FSEF = ReLU(\mathbf{W}_h * Cat(\mathbf{F}_{m+1}, \mathbf{F}_{m+2}, \cdots, \mathbf{F}_n)) + \mathbf{b}_h). \quad (2)$$

Since FSHF encodes the fine details while FSEF captures more semantic information of image contents, one can directly fuse them to generate defocus blur maps. However, the quality of the results cannot be well guaranteed and there are still many in-focus regions being wrongly detected. This is because the fused FSHF still contains some in-focus details and FSEF also contains some incorrect semantic information. Directly using FSHF and FSEF not only provides wrong guidance for defocus blur region detection, but also harms the useful information originally contained in individual layers. To this end, we propose to recurrently fuse and refine the layer-wise features in a cross-layer manner.

In order to leverage the complementary advantages of both shallow layers and deep layers, we aggregate FSHF to each individual deep layer and aggregate FSEF to each individual shallow layer. In such a cross-layer manner, the features extracted by each layer can be refine step by step. Specifically, since the features of shallow layers focus on the fine detail information but lack of semantic information of defocus blur regions, the FSEF can be used to help them better locate semantic defocus blur regions. Similarly, as the features of deep layers capture semantic information but lack of fine details, the FSHF can be used to promote the fine details preservation. In the recurrent aggregation process, the refined feature maps from shallow layers and deep layers are fused again to generate refined FSHF and FSEF, respectively. Then the refined FSHF and FSEF are aggregated respectively to feature maps from shallow layers and deep layers in the next recurrent step.

In order to select the useful multi-level information with respect to the features of each individual layer and reduce the number of feature channels to the original number before next aggregation, we add a convolutional layer for the aggregated feature maps of each layer. The refined feature maps of each layer at the j-th recurrent step can be formulated as follows:

$$\mathbf{F}_i^j = \begin{cases} ReLU(\mathbf{W}_i^j * Cat(\mathbf{F}_i^{j-1}, FSHF^j) + \mathbf{b}_i^j) & i = m+1, \cdots, n \\ ReLU(\mathbf{W}_i^j * Cat(\mathbf{F}_i^{j-1}, FSEF^j) + \mathbf{b}_i^j) & i = 1, \cdots, m \end{cases} \tag{3}$$

where $\mathbf{F}_i^j$ represents the feature maps for the i-th layer at the j-th recurrent step. $FSEF^j$ and $FSHF^j$ represent the FSEF and FSHF at the j-th recurrent step, respectively. $\mathbf{W}_i^j$ and $\mathbf{b}_i^j$ represent the convolutional kernel and bias of the i-th layer at the j-th recurrent step.

3.2. Defocus Maps Fusing

Since the degree of defocus blur is sensitive to image scales, we need to capture multi-scale information for improving final defocus blur detection results. In [48], Zhao et al. proposed to use a multi-stream strategy to fuse the detection results from different image scales. However, this inevitably increase the computational burden of the whole network. In this work, by considering that different layers just extract features of original image in different scales, we impose a supervision signal to each layer by using the deeply supervised mechanism at each recurrent step, then the output score maps of all the layers at the last step are fused to generate the final defocus blur map.

Specifically, we first concatenate the defocus blur maps predicted from n different layers, then a convolution layer is imposed on the concatenated maps to obtain the final output defocus blur map $\mathbf{B}$, which can be formulated as:

$$\mathbf{B} = ReLU(\mathbf{W}_B * Cat(\mathbf{B}_1^t, \mathbf{B}_2^t, \cdots, \mathbf{B}_n^t) + \mathbf{b}_B), \tag{4}$$

where t denotes the last recurrent step; $\mathbf{B}_i^t$ denotes the predicted defocus blur map from the i-th layer at the t-th step; $\mathbf{W}_B$ and $\mathbf{b}_B$ are the weight and bias of the convolution layer on the concatenated defocus blur maps to learn the relationship among these maps. Note that Hu et al. [7] used a similar manner to aggregate deep features for saliency detection, but they did not distinguish features of shallow layers and deep layers.

3.3. Model Training and Testing

Our network uses the VGG [28] as backbone and we implement it by Caffe [8]. We use conv1_2, conv2_2, conv3_3, conv4_3, conv5_3 and pool5 of the VGG network to represent the features of each individual layer, i.e., $n = 6$ in DeFusionNET. The first three layers are regarded as shallow layers, and the rest ones are set as deep layers, i.e., $m = 3$. In addition, in order to enhance the discrimination capability of feature maps at each layer, two more convolutional layers are appended. More details will be found in the released code.

Training: The cross-entropy loss is used for each output of this network during the training process. For the i-th layer at the j-th recurrent step, the pixel-wise cross entropy loss between $\mathbf{B}_i^j$ and the ground truth blur mask $\mathbf{G}$ is calculated as:

$$L_i^j(\boldsymbol{\theta}) = -\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{l\in\{0,1\}} \begin{Bmatrix} \log \Pr(\mathbf{B}_i^j(x,y)=l|\boldsymbol{\theta}) \\ \cdot \mathbf{1}(\mathbf{G}(x,y)=l) \end{Bmatrix} \tag{5}$$

where $\mathbf{1}(\cdot)$ is the indicator function. The notation $l \in \{0, 1\}$ indicates the out-of-focus or in-focus label of the pixel at location (x, y) and $Pr(\mathbf{B}_i^j(x, y) = l|\boldsymbol{\theta})$ represents its corresponding probability of being predicted as blurry pixel or not. $\boldsymbol{\theta}$ denotes the parameters of all network layers.

Based on Eq. (5), the final loss function is defined as the loss summation of all immediate predictions:

$$L = \lambda_f L_f + \sum_{i=1}^{n}\sum_{j=1}^{t} \lambda_i^j L_i^j(\boldsymbol{\theta}), \tag{6}$$

where L_f is loss for the final fusion layer; L_f is the weight for the fusion layer and λ_i^j represents the weight of the i-th layer at the j-th recurrent step. In our experiment, we empirically set all the weights to 1.

Our model is initialized by the pre-trained VGG-16 model and fine tuned on part of Shi et al.'s public blurred image dataset [25], which consists of 1000 blurred images and their manually annotated ground truths. 704 of these images are partially defocus blurred and the rest 296 ones are motion blurred. We divide the 704 defocus blurred images into two parts, i.e., 604 for training and the remaining 100 ones for testing. Since the number of training images is not enough to train a deep neural network, we perform data augmentation by randomly rotating, resizing and horizontally flipping all of the images and their corresponding ground truths, and finally the training set is enlarged to 9,664 images. We train our model on a machine equipped with an Intel 3.4GHz CPU with 128G memory and 2 GPUs (one Nvidia GTX 1080Ti and one Nvidia GTX Titan Xp). We optimize the whole network by using Stochastic gradient descent (SGD) algorithm with the momentum of 0.9 and the weight decay of 0.0005. The learning rate is initially set to 1e-8 and reduced by a factor of 0.1 at 5k iterations. The training batch size is set to 4 and the whole learning process stops after 10k iterations. The training process is completed after approximately 11.6 hours.

Inference: In the testing phase, for each input image, we feed it into our network and obtain the final defocus blur map. Only approximately 0.056s is needed for generating the final defocus blur map for a testing image with 320×320 pixels by using a single Nvidia GTX Titan Xp GPU, which is very efficient.

4. Experiments

4.1. Datasets

In our experiments, two datasets are used for evaluating the performance of our proposed network.

Shi et al.'s dataset [25] contains the rest 100 defocus blurred images as mentioned above.

DUT [48] is a new defocus blur detection dataset which consists of 500 images with pixel-wise annotations. This dataset is very challenging since numerous images contain homogeneous regions, low contrast focal regions and background clutter.

4.2. Evaluation Metrics

Four widely-used metrics are used to quantitatively evaluate the performance of the proposed model: precision-recall (PR) curves, F-measure curves, F-measure scores (F_β) and mean absolute error (MAE) scores. As an overall performance measurement, the F-measure is defined as: $F_\beta = \frac{(1+\beta^2)\cdot precision\cdot recall}{\beta^2\cdot precision+recall}$, where β^2 is set to 0.3 to emphasize precision.The MAE score calculates the average difference between the detected defocus blur map $\mathbf{B}$ and the ground truth $\mathbf{G}$, it is computed as: $MAE = \frac{1}{W\times H}\sum_{x=1}^{W}\sum_{y=1}^{H}|\mathbf{B}(x,y)-\mathbf{G}(x,y)|$, where H and W are the height and width of the input image, respectively.

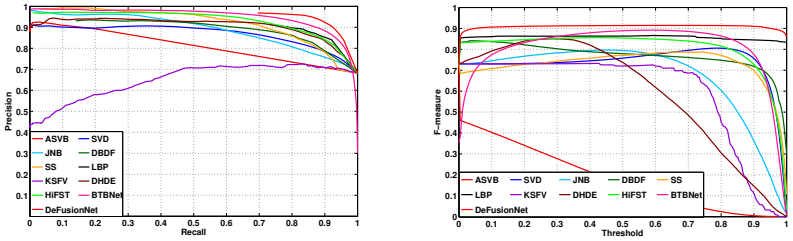
4.3. Comparison with the state-of-the-art methods

We compare our method against other 10 state-of-the-art algorithms, including 2 deep learning-based methods, i..e, multi-scale deep and hand-crafted features for defocus estimation (DHDE) [21] and multi-stream bottom-top-bottom fully convolutional network (BTBNet) [48], and 8 classic defocus blur detection methods, including analyzing spatially-varying blur (ASVB) [3], Singular Value Decomposition based blur detection (SVD) [29], just noticeable defocus blur detection (JNB) [26], discriminative blur detection features (DBDF) [25], spectral and spatial approach (SS) [35], local binary patterns (LBP) [41], classifying discriminative features (KSFV) [20] and high-frequency multi-scale fusion and sort transform of gradient magnitudes (HiFST) [1]. For all of these methods except BTBNet, we use the authors' original implementations with recommended parameters. As to BTBNet, we directly download the results from the authors' project page since they have not released their implementation.

Quantitative Comparison. Table 1 presents the compared results of MAE and F-measure scores. It is observed that our method consistently performs favorably against other methods on the two datasets, which indicates the superiority of our method over other approaches. In Figure 4 and Figure 5, we plot the PR curves and F-measure curves of different methods on different datasets. From the results, we observe that our method also consistently outperforms other counterparts.

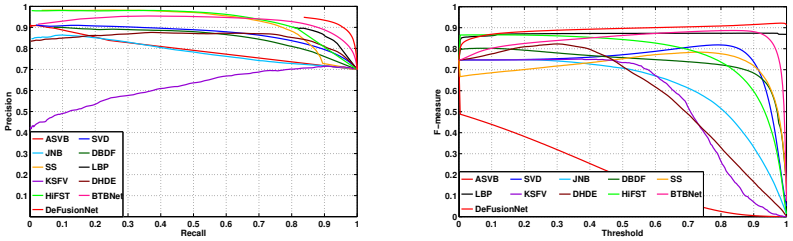
Qualitative Comparison. Figure 6 shows a visual comparison of our method and other ones. As can be seen, our method generates more accurate defocus blur maps when the input image contains in-focus smooth regions and background clutter. In addition, the boundary information of the in-focus objects can be well preserved in our results. More visual comparison results can be found in the supplementary file.

Running Efficiency Comparison. In addition to the appealing results, our proposed DeFusionNet is also efficient for both training and testing. The whole training process of our DeFusionNet takes only about 11.6 hours. As to the testing phase, we use only one GPU (Nvidia GTX Titan Xp). The average running time for an image of different methods on the two different datasets are shown in Table 2. As can be seen, when our DeFusionNet is well trained, it is faster than all of other methods for detecting the defocus



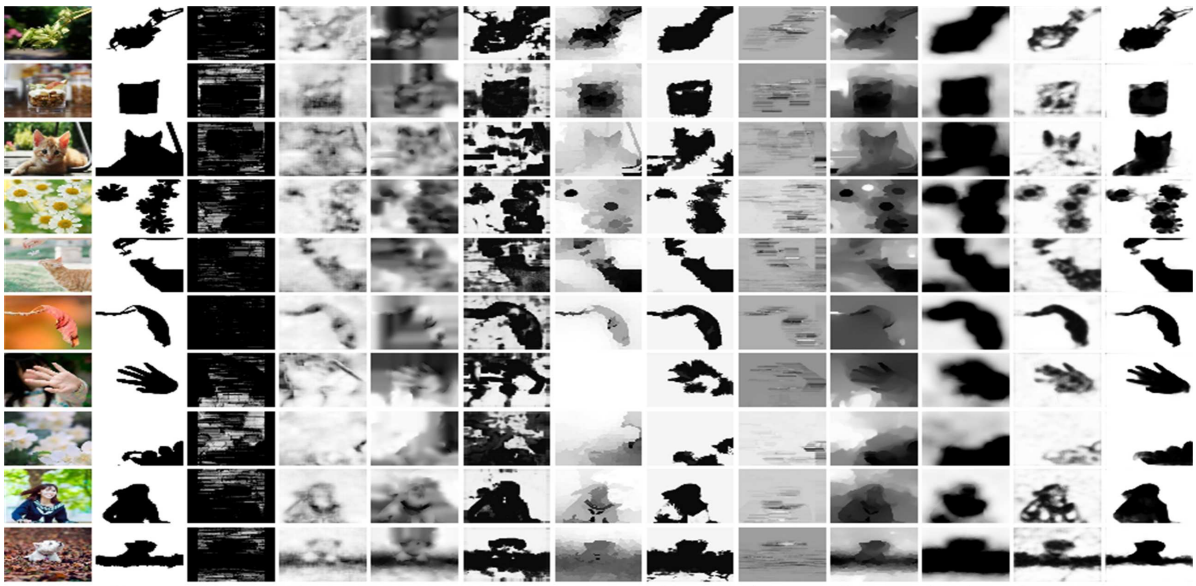
(a) Precision-recall curves (b) F-measure curves

Figure 4. Comparison of precision-recall curves and F-measure curves of different methods on Shi et al.'s dataset.



(a) Precision-recall curves (b) F-measure curves

Figure 5. Comparison of precision-recall curves and F-measure curves of different methods on DUT dataset.



(a) Input (b) GT (c) ASVB (d) SVD (e) JNB (f) DBDF (g) SS (h) LBP (i) KSFV (j) DHDE (k) HiFST (l) BTBNet (m) Ours

Figure 6. Visual comparison of detected defocus blur maps generated from different methods. The results demonstrate that our method consistently outperforms other approaches, and produces defocus blur maps more close to the ground truth.

blur regions from an input image. As to BTBNet, although we cannot evaluate its running time since we do not have its implementation, the authors claimed in their paper that nearly 5 days needed for training BTBNet and approximately 25s is needed to generate the defocus blur map for a testing image with 320×320 pixels. By contrast, the training and testing phases of our DeFusionNet is more efficient.

4.4. Ablation Analysis

Effectiveness of FFRM. In order to validate the efficacy of FFRM, we change the network by fusing the feature maps from all of layers to one group at each recurrent step, then the fused features are used to refine the features of each layer. We denote this network as DeFusionNet_noFFRM for comparison. The F-measure and MAE scores on the t-

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

Datasets	Metric	ASVB	SVD	JNB	DBDF	SS	LBP	KSFV	DHDE	HiFST	BTBNet	DeFusionNet
Shi et al.'s dataset	F_β	0.731	0.806	0.797	0.841	0.787	0.866	0.733	0.850	0.856	0.892	0.917
	MAE	0.636	0.301	0.355	0.323	0.298	0.186	0.380	0.390	0.232	0.105	0.116
DUT	F_β	0.747	0.818	0.748	0.802	0.784	0.874	0.751	0.823	0.866	0.887	0.922
	MAE	0.651	0.301	0.424	0.369	0.296	0.173	0.399	0.408	0.302	0.190	0.115

Table 2. Average running time (seconds) for an image of different methods on different datasets.

Methods		ASVB	SVD	JNB	DBDF	SS	LBP	KSFV	DHDE	HiFST	BTBNet	DeFusionNet
Datasets	Shi et al.'s dataset	2.04	21.09	11.47	214.83	2.76	57.34	32.748	47.06	2576.24	–	0.094
	DUT	1.59	10.91	5.12	110.37	1.20	30.38	20.139	21.51	1169.57	–	0.056

wo datasets are shown in Table 3, and the precision-recall curves are shown in the supplementary. As can be seen, our DeFusionNet with FFRM module performs better than DeFusionNet_noFFRM, which demonstrates that the cross-layer feature fusion manner can effectively capture the complementary information between shallow features and deep semantic features for improving the final results. In addition, DeFusionNet_noFFRM also performs better than other previous methods, this also validates the efficacy of our proposed network structure.

Effectiveness of the Final Defocus Maps Fusion. By considering that the degree of defocus in an image is sensitive to image scales, we fuse the output of different layers at the last recurrent step to form the final result. We also perform ablation experiments to evaluate the effectiveness of the final fusing step. The final outputs of all the layers are represented as DeFusionNet_O1, DeFusionNet_O2, DeFusionNet_O3, DeFusionNet_O4, DeFusionNet_O5, DeFusionNet_O6. We also show the F-measure, MAE scores in Table 3 and the precision-recall curves of these outputs in the supplementary. It can be seen that the fusing mechanism effectively improves the final results.

Effectiveness of the Times of Recurrent Steps. In our DeFusionNet, we fuse and refine the features of each layer in a recurrent and cross-layer manner, the feature maps can be improved step by step. In order to validate whether the features can be improved in a recurrent manner, we report the F-measure and MAE scores by using different times of recurrent step in Table 4. As can be seen from Table 4, the more times of recurrent step, the better results can be obtained. In addition, it should be noted that DeFusionNet can obtain relatively stable results when the times of recurrent is 3. Therefore, we empirically set 3 times of recurrent step in our experiments for the tradeoff between effectiveness and efficiency.

Table 3. Ablation analysis using F-measure and MAE scores.

Methods	Shi et al.'s dataset		DUT	
	F_β	MAE	F_β	MAE
DeFusionNet_noFFRM	0.907	0.154	0.904	0.155
DeFusionNet_O1	0.914	0.118	0.915	0.118
DeFusionNet_O2	0.914	0.118	0.915	0.118
DeFusionNet_O3	0.914	0.118	0.918	0.118
DeFusionNet_O4	0.911	0.127	0.915	0.125
DeFusionNet_O5	0.915	0.118	0.919	0.117
DeFusionNet_O6	0.915	0.117	0.920	0.117
DeFusionNet	0.917	0.116	0.922	0.115

Table 4. Ablation analysis of the times of recurrent steps (DeFusionNet_Step_k represents using k times of recurrent steps in DeFusionNet).

Methods	Shi et al.'s dataset		DUT	
	F_β	MAE	F_β	MAE
DeFusionNet_Step_1	0.702	0.253	0.756	0.321
DeFusionNet_Step_2	0.883	0.132	0.893	0.134
DeFusionNet_Step_3	0.917	0.116	0.922	0.115
DeFusionNet_Step_4	0.918	0.116	0.923	0.115
DeFusionNet_Step_5	0.918	0.115	0.924	0.116
DeFusionNet_Step_6	0.919	0.115	0.924	0.116

5. Conclusions

In this work, we propose a deep convolutional network (DeFusionNet) for efficient and accurate defocus blur detection. Firstly, DeFusionNet combines both shallow-layer features and deep-layer features for generating the final high-resolution defocus blur maps. Secondly, DeFusionNet fuses and refines the features from different players in a cross-layer manner, which can effectively capture the complementary information between shallow features and deep semantics features. Finally, DeFusionNet obtains the final accurate defocus blur map by fusing the outputs from all the layers. Extensive experimental results demonstrate that the proposed DeFusionNet consistently outperforms other state-of-the-art methods in terms of both accuracy and efficiency.

6. Acknowledgments

The work was supported by the National Natural Science Foundation of China (NO. 61701451 and 61773392) and the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) NO. CUG170654. We would also like to thank NVIDIA Corporation for the donation of a Titan Xp GPU card used for this research. Xinzhong Zhu and Xinwang Liu are the corresponding authors of this paper.

References

[1] S Alireza Golestaneh and Lina 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. 1, 3, 6

[2] Soonmin Bae and Frédo Durand. Defocus magnification. *Computer Graphics Forum*, 26(3):571–579, 2007. 1

[3] Ayan Chakrabarti, Todd Zickler, and William T. Freeman. Analyzing spatially-varying blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2512–2519, 2010. 6

[4] Florent Couzinie-Devy, Jian Sun, Karteek Alahari, and Jean Ponce. Learning to estimate and remove non-uniform image blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1075–1082, 2013. 2

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016. 2

[6] James H Elder and Steven W Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, 1998. 2

[7] Xiaowei Hu, Lei Zhu, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng. Recurrently aggregating deep features for salient object detection. In *AAAI*, pages 6943–6950, 2018. 5

[8] Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, and Jonathan. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678, 2014. 5

[9] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *Proceedings of the IEEE international conference on computer vision*, pages 1976–1983, 2013. 1

[10] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017. 2

[11] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 817–825, 2016. 2, 3

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2, 4

[13] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018. 2, 3

[14] Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):2999–3007, 2017. 2

[15] Renting Liu, Zhaorong Li, and Jiaya Jia. Image partial blur detection and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[17] Belen Masia, Adrian Corrales, Lara Presa, and Diego Gutierrez. Coded apertures for defocus deblurring. In *Symposium Iberoamericano de Computacion Grafica*, volume 5, 2011. 1

[18] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 2

[19] Y. Pang, H. Zhu, X. Li, and X. Li. Classifying discriminative features for blur detection. *IEEE Transactions on Cybernetics*, 46(10):2220–2227, 2015. 1, 2

[20] Yanwei Pang, Hailong Zhu, Xinyu Li, and Xuelong Li. Classifying discriminative features for blur detection. *IEEE Transactions on Cybernetics*, 46(10):2220–2227, 2016. 6

[21] Jinsun Park, Yu Wing Tai, Donghyeon Cho, and In So Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2760–2769, 2017. 1, 2, 3, 6

[22] Kuldeep Purohit, Anshul B Shah, and AN Rajagopalan. Learning based single image blur detection and segmentation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2202–2206. IEEE, 2018. 3

[23] Yuankai Qi, Shengping Zhang, Lei Qin, Qingming Huang, Hongxun Yao, Jongwoo Lim, and Ming-Hsuan Yang. Hedging deep features for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[24] E Saad and K Hirakawa. Defocus blur-invariant scale-space feature extractions. *IEEE Transactions on Image Processing*, 25(7):3141–3156, 2016. 1

[25] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2972, 2014. 1, 2, 6

[26] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–665, 2015. 3, 6

[27] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *IEEE Conference on computer vision and pattern recognition*, pages 3118–3126, 2018. 2, 3

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Representation Learning*, 2015. 2, 3, 5

[29] Bolan Su, Shijian Lu, and Chew Lim Tan. Blurred image region detection and classification. In *ACM International Conference on Multimedia*, pages 1397–1400, 2011. 1, 2, 6

[30] Chong Sun, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-aware regressions for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8962–8970, 2018. 2

[31] Yu-Wing Tai and Michael S Brown. Single image defocus map estimation using local contrast prior. In *IEEE Interna-*

tional Conference on Image Processing, pages 1797–1800. IEEE, 2009. 2

[32] Chang Tang, Chunping Hou, Yonghong Hou, Pichao Wang, and Wanqing Li. An effective edge-preserving smoothing method for image manipulation. *Digital Signal Processing*, 63:10–24, 2017. 1, 2

[33] Chang Tang, Chunping Hou, and Zhanjie Song. Defocus map estimation from a single image via spectrum contrast. *Optics letters*, 38(10):1706–1708, 2013. 1, 2

[34] Chang Tang, Pichao Wang, Changqing Zhang, and Wanqing Li. Salient object detection via weighted low rank matrix recovery. *IEEE Signal Processing Letters*, 24(4):490–494, 2017. 1

[35] Chang Tang, Jin Wu, Yonghong Hou, Pichao Wang, and Wanqing Li. A spectral and spatial approach of coarse-to-fine blurred image region detection. *IEEE Signal Processing Letters*, 23(11):1652–1656, 2016. 1, 6

[36] Chang Tang, Jin Wu, Yonghong Hou, Pichao Wang, and Wanqing Li. A spectral and spatial approach of coarse-to-fine blurred image region detection. *IEEE Signal Processing Letters*, 23(11):1652–1656, 2016. 3

[37] Cuong T Vu, Thien D Phan, and Damon M Chandler. s_3: A spectral and spatial measure of local perceived sharpness in natural images. *IEEE Transactions on Image Processing*, 21(3):934, 2012. 1

[38] Xin Wang, Baofeng Tian, Chao Liang, and Dongcheng Shi. Blind image quality assessment for measuring image blur. In *Congress on Image and Signal Processing*, pages 467–470. IEEE, 2008. 1

[39] Guodong Xu, Yuhui Quan, and Hui Ji. Estimating defocus blur via rank of local patches. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Venice, Italy*, pages 22–29, 2017. 3

[40] Ruomei Yan and Ling Shao. Blind image blur estimation via deep learning. *IEEE Transactions on Image Processing*, 25(4):1910–1921, 2016. 2

[41] Xin Yi and Mark Eramian. Lbp-based segmentation of defocus blur. *IEEE Transactions on Image Processing*, 25(4):1626–1638, 2016. 1, 3, 6

[42] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2

[43] Shanghang Zhang, Xiaohui Shen, Zhe Lin, Radomır Mech, Joao P Costeira, and José MF Moura. Learning to understand image blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6586–6595, 2018. 3

[44] Wei Zhang and Wai-Kuen Cham. Single image focus editing. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1947–1954. IEEE, 2009. 1

[45] Wei Zhang and Wai-Kuen Cham. Single-image refocusing and defocusing. *IEEE Transactions on Image Processing*, 21(2):873–882, 2012. 1

[46] Yi Zhang and Keigo Hirakawa. Blur processing using double discrete wavelet transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1098, 2013. 1

[47] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2, 3

[48] Wenda Zhao, Fan Zhao, Dong Wang, and Huchuan Lu. Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3080–3088, 2018. 1, 2, 3, 5, 6

[49] 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. 1

[50] Shaojie Zhuo and Terence Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011. 1, 2