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

Chang Tang, *Member, IEEE,* Xinwang Liu, *Senior Member, IEEE,* Xiao Zheng, Wanqing Li, *Senior Member, IEEE,* Jian Xiong, *Member, IEEE,* Lizhe Wang, *Senior Member, IEEE,* Albert Zomaya, *Fellow, IEEE,*Antonella Longo

Abstract—Albeit great success has been achieved in image defocus blur detection, there are still several unsolved challenges, e.g., interference of background clutter, scale sensitivity and missing boundary details of 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 first fuse the features from different layers of FCN as shallow features and semantic features, respectively. Then, the fused shallow features are propagated to deep layers for refining the details of detected defocus blur regions, and the fused semantic features are propagated to shallow layers to assist in better locating blur regions. The fusion and refinement are carried out recurrently. In order to narrow the gap between low-level and high-level features, we embed a feature adaptation module before feature propagating to exploit the complementary information as well as reduce the contradictory response of different feature layers. Since different feature channels are with different extents of discrimination for detecting blur regions, we design a channel attention module to select discriminative features for feature refinement. Finally, the output of each layer at last recurrent step are fused to obtain the final result. We collect a new dataset consists of various challenging images and their pixel-wise annotations for promoting further study. Extensive experiments on two commonly used datasets and our newly collected one are conducted to demonstrate both the efficacy and efficiency of DeFusionNet.

Index Terms—Defocus Blur Detection, Multi-scale Features, Feature Fusing, Channel Attention.

1 Introduction

As a common phenomenon, defocus blur occurs when objects in a scene are not within the camera's depth of focus. Defocus blur detection, which aims to detect the out-of-focus regions from an image, has gained much attention due to its wide range of potential applications such as image quality assessment [36], [42], salient object detection [10], [38], image deblurring [21], [29], defocus magnification [2], [37] and image refocusing [50], [51], just list a few.

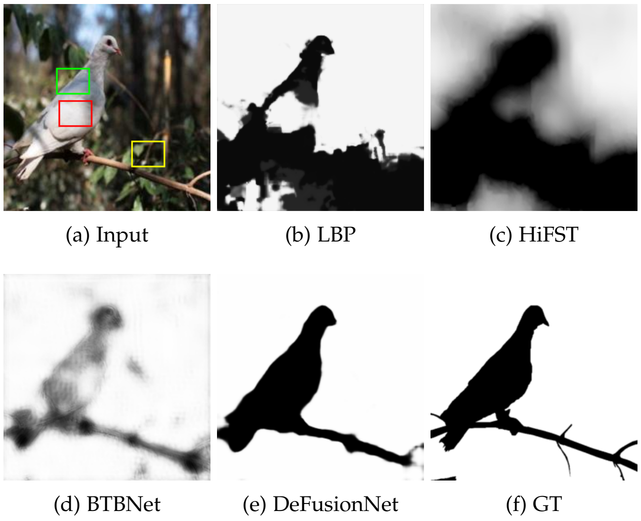

(a) Input (b) LBP (c) HiFST
(d) BTBNet (e) DeFusionNet (f) GT

Fig. 1. Some challenging cases for defocus blur detection. (a) Input image, defocus blur detection map obtained by (b) LBP [46], (c) HiFST [1], (d) BTBNet [56], (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 generally 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 to model the edge changes since defocus blur usually blunts object edges in an image [18], [23], [25], [28],

- *Chang Tang and Lizhe Wang are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China. E-mail: tangchang@cug.edu.cn and Lizhe.Wang@gmail.com.*
- *Xinwang Liu and Xiao Zheng are with the School of Computer Science, National University of Defense Technology, Changsha 410073, China. Xinwang Liu is also with Peng Cheng Laboratory, Shenzhen, China. E-mail: xinwangliu@nudt.edu.cn and endozheng@gmail.com.*
- *Wanqing Li is with the School of Computing and Information Technology, University of Wollongong, NSW, 2500, Australia. E-mail: wanqing@uow.edu.au.*
- *J. Xiong is with School of Business Administration, Southwestern University of Finance and Economics, Chengdu, Sichuan, 611130, China. E-mail: xiongjian2017@swufe.edu.cn.*
- *Albert Zomaya is with the School of Information Technologies, University of Sydney, NSW 2006, Australia. E-mail: albert.zomaya@sydney.edu.au.*
- *Antonella Longo is with the Department of Innovation Engineering, University of Salento, Lecce, Italy. E-mail: antonella.longo@unisalento.it.*

Manuscript received February xx, 2020; revised xx xx, 2020. This work was supported by the National Natural Science Foundation of China (No. 61701451, 61773392, 61901205, U1711266 and 41925007) and the Opening Fund of Key Laboratory of Geological Survey and Evaluation of Ministry of Education (No. GLAB2020ZR18) and the Fundamental Research Funds for the Central Universities. (Xinwang Liu and Xiao Zheng are the corresponding authors.)

[29], [33], [39], [41], [52], [59], [60]. Although great advances have been made by using traditional hand-craft features, these methods are affected by a number of challenges. First, traditional low-level features cannot distinguish well blurred regions which do not contain structural information from in-focus smooth regions. Second, these methods do not utilize global semantic information which is critical for detecting low-contrast focal regions (as shown in the red rectangular region of Figure 1a) and dealing with cluttered background (as shown in the yellow rectangular region of Figure 1a). In addition, the edge information between in-focus regions and blurry regions has not been well preserved (as shown in the green rectangular region of Figure 1a).

Recently, due to their strong feature extraction and learning capability, deep convolutional neural networks (CNNs) have made remarkable advances in various computer vision tasks, such as image classification [14], [32], object detection [12], [17], object tracking [16], [27], [34], scene semantic segmentation [19], [22], [54], image de-noising [11], [48] and super-resolution [4], [31]. As a result, CNNs have been used for the detection of image defocus blur regions. In [45], a pre-trained deep neural network and a general regression neural network are proposed to classify the types of blurring and then estimate their parameters. By systematically analyzing the effectiveness of different defocus detection features, Park et al. [25] 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 content. In [56], 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 sensitivity of defocus degree to image scales. Although significant improvement has been attained by BTBNet, it uses a forward stream and a backward stream to integrate features from different levels at each image scale, this leads to high computational complexity for both network training and testing, and the complementary information of different layers cannot been fully exploited. Consequently, cluttered background cannot be dealt with properly. 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 discriminative deep features across deep and shallow layers in an alternate and cross-layer manner, then the complementary information of features from different layers can be fully exploited for boosting defocus blur detection performance.

This manuscript is a significant extension of the conference version [40], and it differs [40] with following additional contributions:

- Compared with the conference version, we newly designed a channel attention module and integrated it into the DeFusionNET for selecting discriminative features to further boost the feature refining process.
- Considering that most of previous deep neural networks mainly integrate multiple level deep features indiscriminately by commonly used operations such as addition, concatenation and multiplication while ignore the gap between different feature layers, we introduce a feature adaptation module and embed it into our network before feature propagating, which is designed to exploit the complementary information as well as reduce the contradictory response of different layers.
- More experiments were conducted with new evaluation criteria to evaluate and analyze the proposed network. Results of the benchmarking methods on different datasets will be publicly released for academic usage.
- A new dataset which consists of 150 challenging images and their corresponding pixel-level annotations was collected. The proposed network has been successfully validated using both previous datasets and our newly collected one. The newly collected dataset will be made publicly available for further academic research and evaluation.

2 Related Work

As a sub-field of computer vision, defocus blur detection has been widely investigated due to its important role in many practical applications. Therefore, a variety of defocus blur detection methods have been put forward, which can be roughly categorized into two classes, i.e., hand-crafted features based methods and deep learning based methods. Following we give a brief review about these methods.

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 [3], [5], [35], [36], [37], [60]. 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. [33] detected blur regions by examining the singular values for each image pixel. Shi et al. [29] studied a series of blur feature representations such as gradient and data-driven local filters features to enhance discriminative power for differentiating blurred and unblurred image regions. In [23], Pang et al. developed a kernel-specific feature to detect blur regions of an image, 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. [30] proposed a simple yet effective blur feature via sparse representation and image decomposition. Yi and Eramian [46] designed a sharpness metric based on local binary patterns and the in- and out-of-focus image regions are separated by using the metric. Since the blur can affect the spectrum of an image, Tang et al. [39] designed a log averaged spectrum residual metric to estimate the blur amount of edge pixels, 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. Xu et al. [44] presented a fast yet effective approach to estimate the spatially varying amounts of defocus blur at edge locations based on maximizing the ranks of the corresponding local patches, then the complete defocus map is generated by a standard propagation procedure.

Hand-crafted feature based methods work well for images with simple structures but are not effective enough for scenes with complex contents. Therefore, extracting high level and more discriminative features are necessary.

2.2 Deep Learning based Methods

Due to their ability in learning to extract hierarchical features, deep CNNs based methods have refreshed the records of many computer vision tasks [12], [16], [31], [32], [54], including defocus blur detection [8], [13], [15], [20], [25], [55], [56], [58]. In [25], high-dimensional deep features are first extracted by using a CNN, then these features and traditional hand-crafted features are concatenated together and fed into a fully connected neural network for determining the degree of defocus. Purohit et al. [26] proposed to train two sub-networks to learn global context and local features respectively, then the pixel-level probabilities estimated by the two networks are aggregated and feed into a MRF based framework for blur region segmentation. Zhang et al. [49] proposed a dilated fully convolutional neural network with pyramid pooling and boundary refinement layers to generate blur response maps. In [20], Ma et al. demonstrated that the high-level semantic information is critical for defocus identification. Considering that the degree of defocus blur is sensitive to scales, Zhao et al. [56] proposed a multi-stream bottom-top-bottom fully convolutional network (BTBNet) to integrate low-level cues with high-level semantic information for defocus blur detection. Feature aggregation [40] and ensemble networks [58] are also proposed for this task. Lee et al. [15] produced a novel depth-of-field dataset with synthetically blur for network training. Although significant improvement has been obtained by existing deep neural networks, there are still several issues which make the detected results not satisfactory enough for some subsequent tasks. First, most of previous deep neural networks directly integrate multiple level deep features by commonly used operations such as addition, concatenation and multiplication, but ignore the gap between different levels of features. Second, the high-level context features which are critical for discriminating in-focus smooth regions may be diluted as they pass on the top-down flow stream. Third but not last, the redundancy existed in the high-level features is not sufficiently suppressed while the channel-wise attention is not well exploited.

In this work, we propose an effective and efficient defocus blur detection deep neural network via recurrently fusing and refining multi-scale discriminative deep features (DeFusionNET). Instead of directly refining the detection score map as many previous deep CNNs based detection methods do, we recurrently fuse and refine the features of different layers in DeFusionNET. Particularly, a feature fusing and refining module (FFRM) is designed to exploit the complementary information of low-level cues and high-level semantic features 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. Considering that directly integrating features from multiple layers could ignore the gap between different feature layers, we introduce a feature adaptation module and embed it into our network to avoid the contradictory response of different layers. Since different scales of receptive views produce the features with different extents of discrimination, we design and integrate a channel attention module after the feature fusing at each step to select more discriminative features to refine the layer-wise features. Note that different layers of a CNN extract features at different scales of an image and the degree of defocus blur is sensitive to image scales, we fuse the detection score maps estimated from different network 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 methods in terms of both accuracy and efficiency.

3 Proposed DeFusionNET

The proposed DeFusionNet 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 DeFusionNET.

For an effective defocus blur detection network, it should require both low-level cues and high-level semantic information for generating 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 the impact of background clutters. In addition, there are often some smooth in-focus regions within an object, the high-level semantic information produced by the deep layers can avoid these regions being detected as blurry regions. Furthermore, since the degree of defocus is sensitive to image scales, the network should be capable of making use of multi-scale features to improve the final results. Finally, the network should be easily fine-tuned because there are often no sufficient labeled defocus blur images for training such a deep network.

Specifically, we choose the VGG network [32] as the backbone feature extraction network and use the pre-trained VGG16 model to initialize the network. Firstly, we use our network to extract a set of hierarchical features which encode the low-level details and high-level semantic information at different scales of an image. On the one hand, since a series of spatial pooling and convolution operations progressively reduce the spatial resolution of the initial image, the fine details of image structure are inevitably gradually lost, which is harmful for densely distinguishing 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 defocus blur regions. Therefore, how to exploit the complementary information extracted from shallow layers and deep layers is critical for the detection of

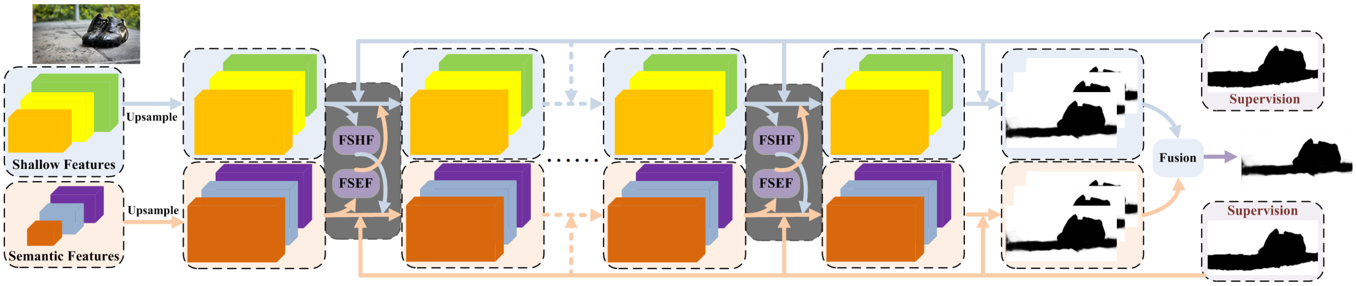


Fig. 2. The pipeline of our DeFusionNET. The dark gray block represents the proposed FFRM module. For a given image, we first extract its multi-scale features by using the basic VGG network. Then the features from shallow layers and deep layers are fused as FSHF and FSEF, respectively. Considering the 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.

defocus regions. As to the low-level and high-level feature maps, they are both upsampled to the same size of the 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 features extracted from different layers are of different spatial scales and the degree of defocus blur is sensitive to image spatial scales, the detection score maps from different layers are fused at the last recurrent step to generate the final defocus blur detection map.

3.1 Feature Fusing and Refining Module (FFRM)

The success of deep CNNs owes to its strong capacity of hierarchically extracting abundant semantic as well as fine details information from different layers. As discussed aforementioned, features from both shallow and deep layers are important for defocus blur region detection. Therefore, we need to 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 due to the noisy and redundant information. 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. In addition, there exist redundancy, complement as well as contradictory response from the features extracted from different layers, refining deep features by using commonly used operations such as addition, concatenation and multiplication could ignore these information between different layers. Therefore, we introduce and embed a feature adaptation module (FAM) before feature propagating to exploit the complementary information as well as reduce the contradictory response of different feature layers.

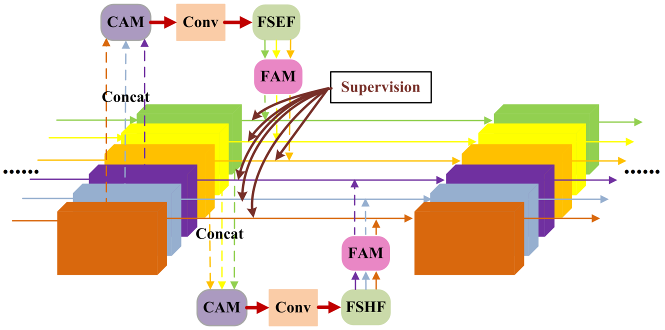


Fig. 3. The architecture of the proposed feature fusing and refining module (FFRM).

Supposing there are n total layers in the network, the first m layers are regarded as shallow layers and the rest ones as deep layers. For the feature maps generated from each shallow layer, we upsample them to the size of the input image by using the deconvolution operation and concatenate them together. Since different scales of receptive views produce the features with different extents of discrimination, a channel attention module (CAM) which will be introduced in the next subsection is added after the concatenated feature maps to select more discriminative features. Then a convolution layer with 1×1 kernel is employed to to the discriminative concatenated feature maps is used to generate FSHF. The FSHF can be mathematically defined as follows:

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

where $\mathbf{F}_i^w \in W \times H \times C$ denotes the weighted 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 that need to be learned during training and $ReLU$ is the ReLU activation function [14].

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

$$FSEF = ReLU(\mathbf{W}_h * Cat(\mathbf{F}^w_{m+1}, \mathbf{F}^w_{m+2}, \cdots, \mathbf{F}^w_n)) + \mathbf{b}_h). \tag{2}$$

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, this strategy would lead many in-focus regions being wrongly detected as defocus regions. This is because the fused FSHF still contains some in-focus details and FSEF also contains some noisy 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 from each layer can be refined 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 provide the needed high-level information for the localization of 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 the 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, a convolutional layer is added to the aggregated feature maps for each layer. The refined feature maps of each layer at the j-th recurrent step can be formulated as follows:

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

where $\mathbf{F}^j_i$ 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. $\mathbf{W}^j_i$ and $\mathbf{b}^j_i$ represent the convolutional kernel and bias of the i-th layer at the j-th recurrent step. In order to narrow the gap between shallow and deep layers, we pass the FSEF and FSHF to a feature adaptation module (FAM) at each recurrent step. The details of FAM will be presented in the Subsection 3.3.

3.2 Channel Attention Module (CAM)

Previous deep learning based defocus blur detection methods [25], [26], [49], [56], [57] ignore the possible bias of different feature channels and regard different feature channels contributing equally to the final result, which is not effective in dealing with various types of information. In Figure 4c and 4e, we show the first 36 channel-wise feature maps of the concatenated low level shallow features and high level semantic features in the first fusing step, respectively. As can be seen, different feature channel contributes significantly differently to the defocus blur detection task. On the one hand, most of the feature channels in Figure 4c capture the fine details of image contents, e.g., the edges of the petals. On the other hand, the feature maps in Figure 4e usually focus on the semantic information of the image, including in-focus areas and the rest blurry parts, which can help discriminate low-contrast in-focus regions. In addition, in both Figure 4c and Figure 4e, there are many features that would contribute little to or even impair the detection.

Therefore, we design a channel attention module (CAM) to learn the weights for adaptively rescaling channel-wise features and integrate the CAM into DeFusionNet to boost the feature refining process. Different to previous channel attention model [47], [53] which only uses global average pooling to capture the global statistics of feature maps, we use both global average pooling (GAP) and global maximum pooling (GMP) to aggregate global information, and design the CAM in a dual manner. Figure 5 briefly presents the architecture of the proposed CAM. Given multiple channel-wise feature maps, we firstly leverage GAP and GMP to convert channel-wise global spatial features into vector descriptors, respectively. In such a manner, the statistics of the vector descriptors can serve to express the whole image [6]. In our CAM, the GAP captures the size of blurry regions, while GMP focuses on the defocus intensity. Denoting $\mathcal{F} = [\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_C]$ as the concatenated feature maps with C channels, and the size of different channels is $W \times H$. Then the channel-wise statistics ($\mathbf{s}_{ga} \in \mathbb{R}^C$ and $\mathbf{s}_{gm} \in \mathbb{R}^C$) obtained through the GAP and the GMP operations. Specifically, the c-th element of $\mathbf{s}_{ga}$ and $\mathbf{s}_{gm}$ can be calculated as:

$$\mathbf{s}^{ga}_c = GAP(\mathbf{F}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{F}_c(i,j), \tag{4}$$

and

$$\mathbf{s}^{gm}_c = GMP(\mathbf{F}_c) = max\{\mathbf{F}_c(i,j)\}. \tag{5}$$

In order to learn the non-linear interactions between different feature channels and non-mututually-exclusive relationship between channels, we merge the two attention vectors using element-wise summation and leverage a simple gating mechanism [6], [53] with a sigmoid function and the final channel weights can be expressed as follows:

$$\mathbf{w} = f(\mathbf{W}_U * (ReLU(\mathbf{W}_D * \mathbf{s}^{ga}) + ReLU(\mathbf{W}_D * \mathbf{s}^{gm}))), \tag{6}$$

where $f(\cdot)$ is the sigmoid gating function. $\mathbf{W}_D$ and $\mathbf{W}_U$ are the convolution coefficients of channel-downscaling layer and channel-upscaling layer, respectively (see Figure 5). Then, the final weighted channel-wise feature maps can be written as:

$$\mathbf{F}^w_c = \mathbf{w}_c \cdot \mathbf{F}_c. \tag{7}$$

In Figure 4d and 4f, we intuitively show the channel weights of the feature maps of Figure 4c and 4e, respectively. As can be seen, the proposed CAM can effectively learn different weights for different feature channels. In Figure 4g and 4h, we show the weighted feature maps of Figure 4c and Figure

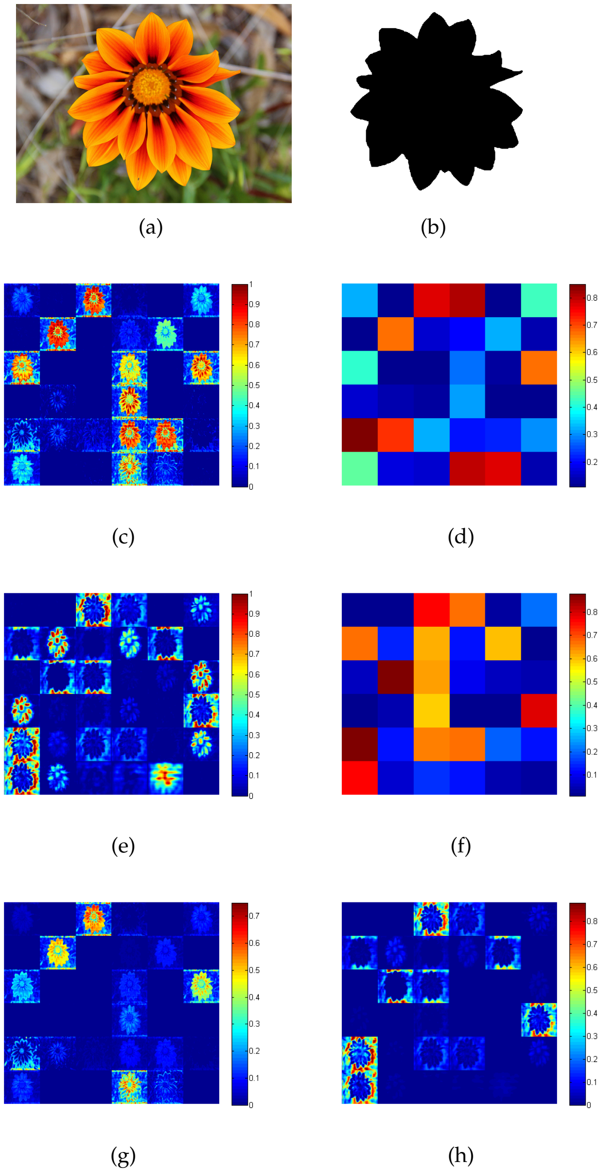


Fig. 4. An intuitive representation of channel-wise feature maps their corresponding channel weights learned by the proposed CAM. (a) Input image, (b) the ground truth of defocus blur detection map, (c) the first 36 channel-wise feature maps of the concatenated low level shallow features in the first fusing step, (d) the corresponding channel weights of the feature maps in (c), (e) the first 36 channel-wise feature maps of the concatenated high level semantic features in the first fusing step, and (f) the corresponding channel weights of the feature maps in (e), (g) and (h) are the weighted feature maps of (c) and (e), respectively.

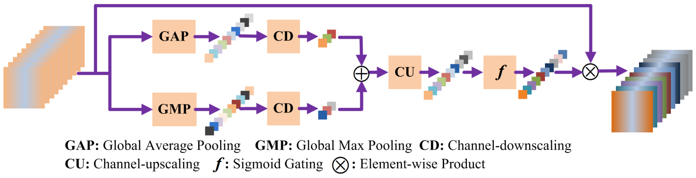

GAP
CD
GMP
CD
CU
f
GAP: Global Average Pooling GMP: Global Max Pooling CD: Channel-downscaling
CU: Channel-upscaling *f*: Sigmoid Gating ⊗: Element-wise Product

Fig. 5. The architecture of the proposed channel attention module (CAM).

4e, which validate that the learned channel weights can strengthen the role of some important feature channels as well as weaken the influence of some useless channels.

3.3 Feature Adaptation Module (FAM)

As done in our previous work [40], the FSEF and FSHF are used to refine the features of shallow layers and deep layers, respectively by directly using concatenation. Since features extracted from shallow layers focus more on fine details of an image while features extracted from deep layers capture more semantic information, directly concatenating them could ignore the gap between different feature layers since the redundancy can not be sufficiently suppressed while the complementary can not be effectively exploited. In addition, there are some contradictory response of different layers, which will dilute the semantic information by adding the FSHF to deep layers, as well as damage the details by adding the FSEF to shallow layers. Therefore, we design a FAM to adjust FSEF and FSHF before feature refining. Since the operation for both FSEF and FSHF is symmetrical, we use the same structure for FSEF and FSHF. Figure 6 briefly presents the architecture of the proposed FAM. The two convolution layers marked in the light green box are used to learn the feature weight of each position, and the FSEF/FSHF are weighted by the learned weight maps. The first convolution layer in the upper left of the light green box is a traditional convolution operation for feature extraction, while the second convolution layer in the lower right of the light green box is used to learn the feature weight of each position. In such manner, the FSEF/FSHF can be re-scaled, i.e., the complementary information between different feature layers can be enhanced, while the contradictory information can be effectively reduced, by using the element-wise production. After that, the adjusted features are added to original FSEF/FSHF for generating the output of FAM, which is used for cross-layer feature refining. The efficacy of FAM will be validated in the experiments section.

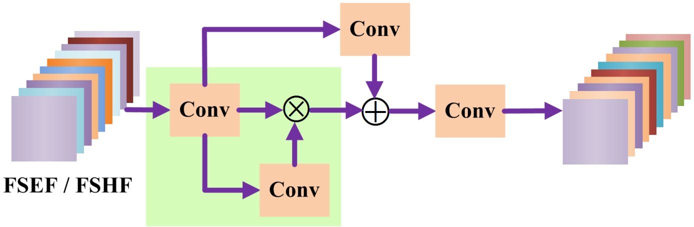


Fig. 6. The architecture of the proposed feature adaptation module (FAM).

3.4 Defocus Maps Fusing

Since the degree of defocus blur is sensitive to image scales, multi-scale information is required for accurate defocus blur detection. In [56], Zhao et al. proposed a multi-stream strategy by fusing 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 the original image in different scales, we impose a supervision signal to each layer by using the deeply supervised mechanism [43] 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 applied on the concatenated maps to obtain the final output defocus blur map $\mathbf{B}$, which can be formulated as:

$$\mathbf{B} = ReLU(\mathbf{W}_B * Cat(\mathbf{B}_1^t, \mathbf{B}_2^t, \cdots, \mathbf{B}_n^t) + \mathbf{b}_B), \tag{8}$$

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.

3.5 Model Training and Testing

Our network uses the VGG [32] as backbone and we implement it by Caffe [9]. 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:

$$L_i^j(\boldsymbol{\theta}) = -\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{l\in\{0,1\}} \begin{Bmatrix} \log \Pr(\mathbf{B}_i^j(x,y)=l|\boldsymbol{\theta}) \\ \cdot \mathbf{1}(\mathbf{G}(x,y)=l) \end{Bmatrix} \tag{9}$$

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. (9), the final loss function is defined as the loss summation of all immediate predictions:

$$L = \lambda_f L_f + \sum_{i=1}^{n}\sum_{j=1}^{t} \lambda_i^j L_i^j(\boldsymbol{\theta}), \tag{10}$$

where L_f is loss for the final fusion layer; λ_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 experiments, 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 [29], 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 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 11.7 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 Titan Xp GPU, which is very efficient.

4 Experiments

4.1 Datasets

As far as we know, there are only two public datasets available for evaluating the performance of pixel-level defocus blur detection algorithms, they are as follows:

Shi et al.'s dataset [29] contains 704 partially defocus blurred images with manually annotated ground truths. Except for the first 604 images of this dataset used for training our network, the rest 100 ones are used for testing.

DUT [56] is a new defocus blur detection dataset which consists of 500 images and their pixel-wise annotations. This is a very challenging dataset since a large number of images contain homogeneous regions, low contrast focal regions and background clutter.

Based on our observations, in most of the images of above mentioned two datasets, the foreground objects are usually in-focus while the background is usually blurry, which leads to the fact that the blur detection methods may be biased to object regions and reduce to foreground/background segmentation. In reality, foreground objects with strong semantic meaning may also be defocused. In addition, the content contained in the images of previous datasets are easy, nearly no complex background or foreground. With these points in mind, we collect a new dataset (referred to as CTCUG) which contains 150 images with manual pixel-wise annotations. We invite five students to manually annotate the defocus areas from each image and the final annotated ground truths are obtained by averaging the results from the five independently labelled masks. In Figure 7, we present some example images and their manually annotated ground truths of our dataset. In the process of collecting our dataset, we take the following challenging settings into consideration:

1) In most of images, the background is in-focus while the foreground regions are blurry (see the first three columns of Figure 7);
2) For some scenes, we take a pair of images with different defocus areas. One of the image is with in-focus background and out-of-focus foreground. The other image is with out-of-focus background and in-focus foreground (see the forth and fifth columns of Figure 7);

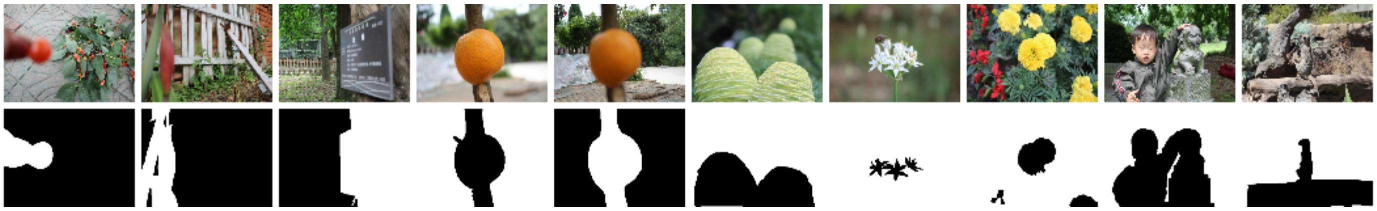


Fig. 7. Some example images and their annotated ground truths of the CTCUG dataset.

3) For the same class of objects, some of them are in-focus while the others are out-of-focus (see the sixth to eighth columns of Figure 7);
4) The images are with complex background and the in-focus area has low contrast (as shown by the last two columns of Figure 7).

These challenges will be validated in the latter subsection. Our newly collected dataset will be made publicly available for further defocus blur detection researches.

4.2 Evaluation Metrics

Six widely-used metrics are used to quantitatively evaluate the performance of the proposed model: precision-recall (PR) curves, F-measure curves, the receiver operating characteristic (ROC) curve, area under the ROC curve (AUC), F-measure scores (F_β) and mean absolute error (MAE) scores. As an overall performance measurement, the F-measure is defined as:

$$F_\beta = \frac{(1+\beta^2)\cdot precision \cdot recall}{\beta^2 \cdot precision + recall}, \tag{11}$$

where β^2 is set to 0.3 to emphasize precision. As neither precision nor recall measure evaluate the true negative saliency assignments, we also use the mean absolute error (MAE) as a complementary. 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:

$$MAE = \frac{1}{W \times H}\sum_{x=1}^{W}\sum_{y=1}^{H}|\mathbf{B}(x,y) - \mathbf{G}(x,y)|, \tag{12}$$

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 11 state-of-the-art algorithms, including 5 deep learning-based methods, i.e., multi-scale deep and hand-crafted features for defocus estimation (DHDE) [25], multi-stream bottom-top-bottom fully convolutional network (BTBNet) [56], deep blur mapping via exploiting high-Level semantics (DBM) [20], defocus map estimation using domain adaptation (DMENet) [15] and our previous DeFusionNet without CAM and FAM (CVPR19) [40], and 6 classic defocus blur detection methods, including just noticeable defocus blur detection (JNB) [30], discriminative blur detection features (DBDF) [29], spectral and spatial approach (SS) [39], local binary patterns (LBP) [46], classifying discriminative features (KSFV) [24] and high-frequency multi-scale fusion and sort transform of gradient magnitudes (HiFST) [1]. For all of these methods, we use the authors' original implementations with recommended parameters.

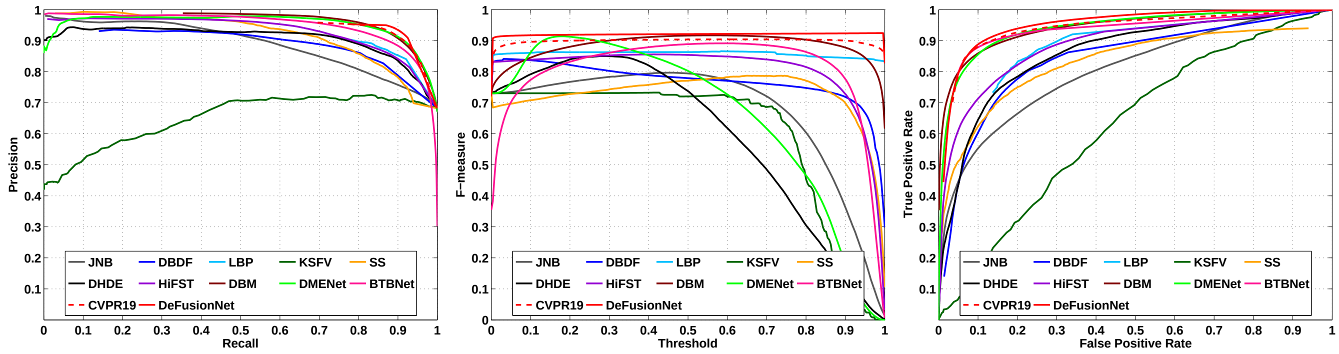
Quantitative Comparison. Table 1 presents the compared results of MAE, F-measure and AUC scores. It is observed that our method consistently performs favorably against other methods on the three datasets, which indicates the superiority of our method over other ones. In Figure 8, Figure 9 and Figure 10, we plot the PR curves, F-measure curves and ROC curves of different methods on different datasets. From the results, we observe that our method also consistently outperforms other counterparts.

Qualitative Comparison. Figure 11 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 boundaries of the in-focus objects can be well preserved in our results. It should be noted that some previous deep neural network based methods such as DBM [20], DMENet [15] and BTBNet [56] can not obtain satisfactory results. Since both DBM and BTBNet rely heavily on high-level semantic information, there results loss a large amount of fine details of region boundaries. DMENet aims to estimate the defocus blur amount of different image regions, therefore, some in-focus regions are wrongly detected as slight blur regions. As to our DeFusionNet, both high-level semantic information and low-level details are fully captured. Therefore, we can obtain better results with complete blur regions. More visual comparison results can be found in the online project page [1].

Running Efficiency. Since the coding languages (Matlab, Python and C++) and running platforms (CPU/GPU) are different among different methods, the directly running timing comparison makes little sense, we only report our running time here. The full training process of the DeFusionNet took only about 11.7 hours. As to the testing phase, only one GPU (Nvidia Titan Xp) was used. The average running time for an image of different methods on the three different datasets are 0.097s, 0.059s and 0.068s, respectively. Note 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, our DeFusionNet is more efficient.

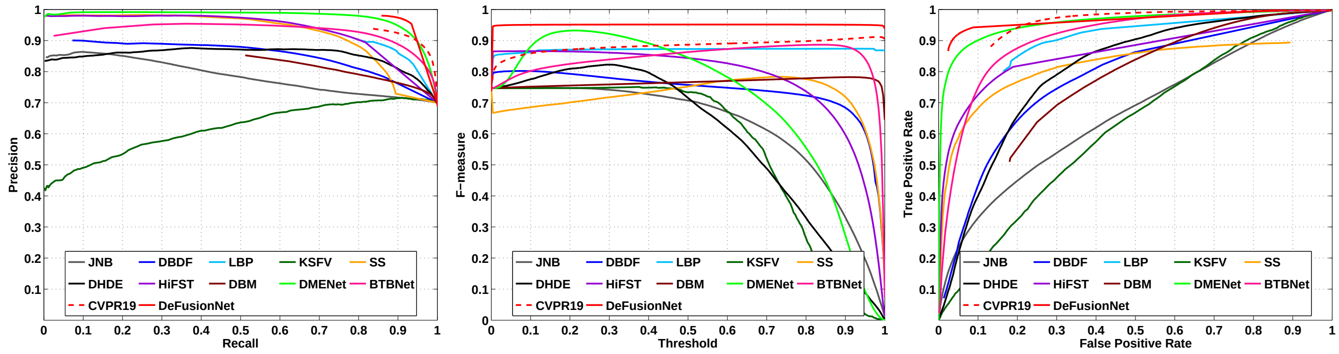
Convergence Property of the Training Process As stated in section 3.5, the whole learning process of the network stops after 10k iterations. In order to validate the convergence property of the whole training process, we plot the training loss with different iteration times in Figure 12. As can be

[1] http://tangchang.net



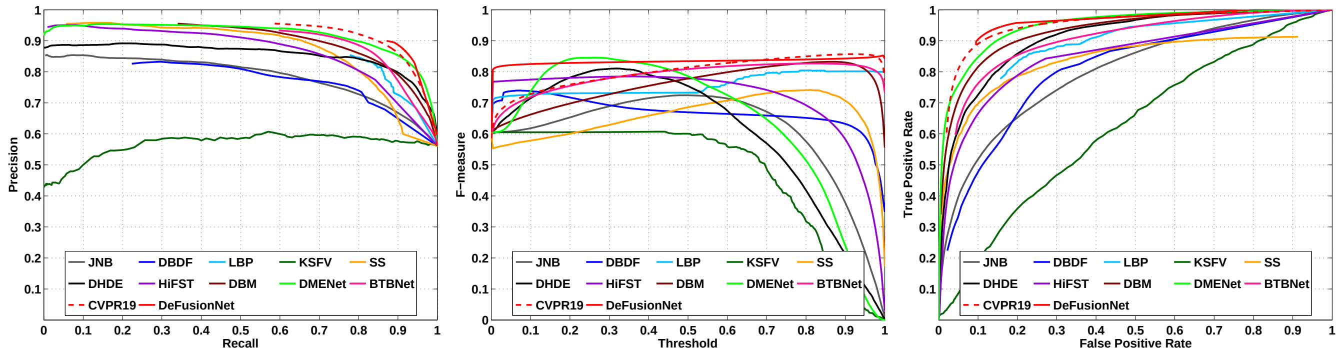
(a) Precision-recall curves (b) F-measure curves (c) ROC curves

Fig. 8. Comparison of precision-recall curves, F-measure curves and ROC curves of different methods on Shi et al.'s dataset.



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

Fig. 9. Comparison of precision-recall curves, F-measure curves and ROC curves of different methods on DUT dataset.



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

Fig. 10. Comparison of precision-recall curves, F-measure curves and ROC curves of different methods on CTCUG dataset.

TABLE 1
Quantitative comparison of F-measure, MAE and AUC scores (The up-arrow ↑ indicates the larger value achieved, the better performance is, while the down-arrow ↓ indicates the smaller, the better).

Datasets	Metric	JNB	DBDF	SS	LBP	KSFV	DHDE	HiFST	DBM	DMENet	BTBNet	CVPR19	DeFusionNet
Shi et al.s dataset	F_β ↑	0.797	0.841	0.787	0.866	0.733	0.850	0.856	0.917	0.914	0.892	0.917	0.925
	MAE↓	0.355	0.323	0.298	0.186	0.380	0.390	0.232	0.155	0.343	0.105	0.116	0.102
	AUC↑	0.594	0.594	0.613	0.603	0.541	0.613	0.619	0.638	0.637	0.831	0.836	0.844
DUT	F_β ↑	0.748	0.802	0.784	0.874	0.751	0.823	0.866	0.782	0.932	0.887	0.922	0.952
	MAE↓	0.424	0.369	0.296	0.173	0.399	0.408	0.302	0.279	0.314	0.190	0.115	0.082
	AUC↑	0.547	0.573	0.607	0.599	0.547	0.592	0.605	0.564	0.635	0.616	0.632	0.643
CTCUG	F_β ↑	0.724	0.740	0.741	0.805	0.607	0.811	0.785	0.832	0.845	0.827	0.891	0.899
	MAE↓	0.347	0.344	0.302	0.242	0.461	0.307	0.267	0.209	0.301	0.177	0.138	0.127
	AUC↑	0.648	0.626	0.664	0.653	0.573	0.680	0.657	0.678	0.694	0.672	0.693	0.705

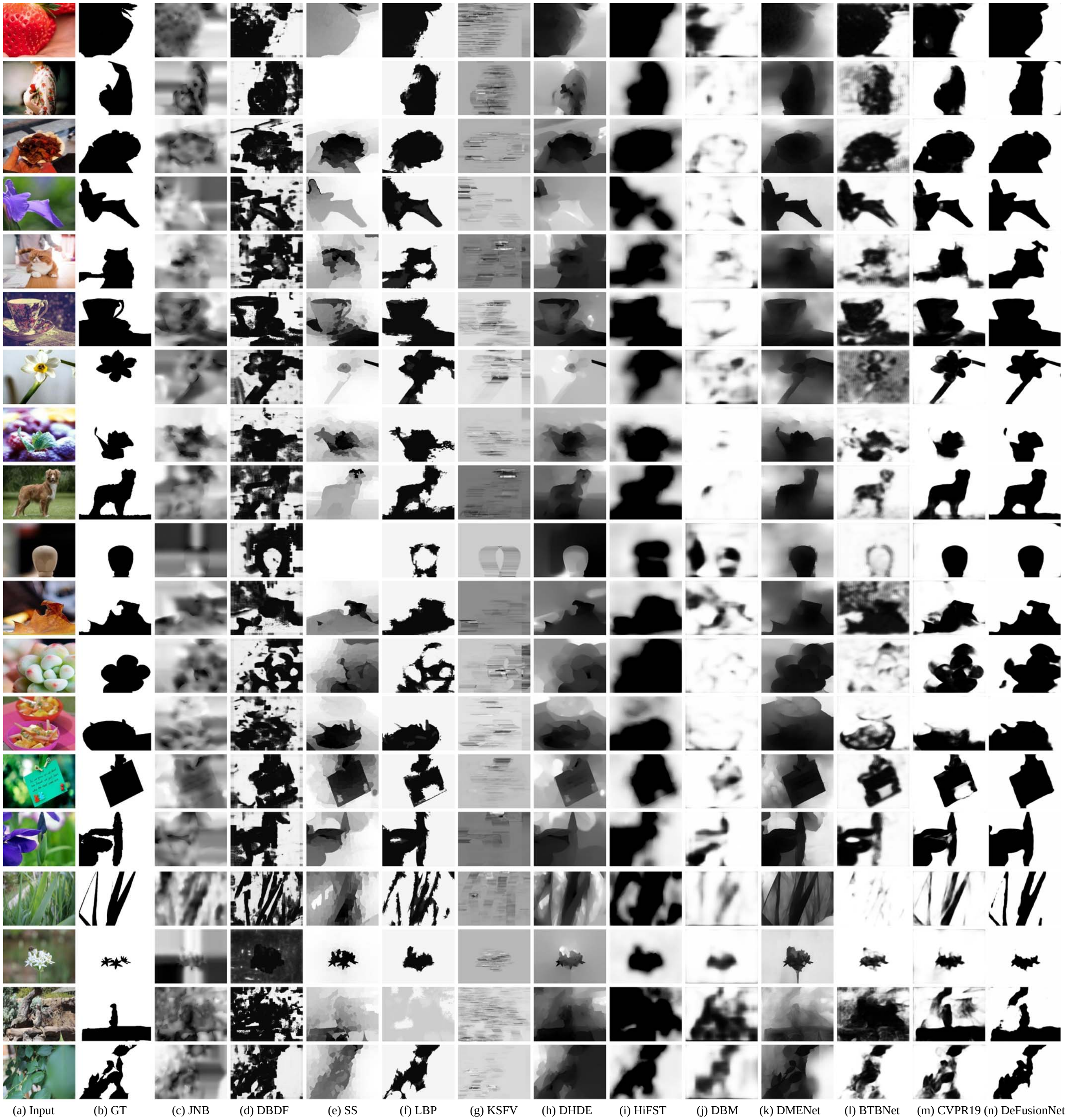


Fig. 11. Visual comparison of detected defocus blur maps generated from different methods. The results demonstrate that our method consistently outperforms other approaches, and produces defocus blur maps more closer to the ground truth.

seen, the training loss goes stable after about 9000 iterations. Therefore, we stop the learning process of the network after 10k iterations for reliable estimation.

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 noFFRM for comparison. The F-measure, MAE and AUC scores on the three datasets are shown in Table 2. As can be seen, our DeFusionNet with FFRM module performs better than 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, noFFRM also performs better than other previous methods, this also validates the efficacy of our proposed network structure.

Effectiveness of CAM and FAM. In order to validate the

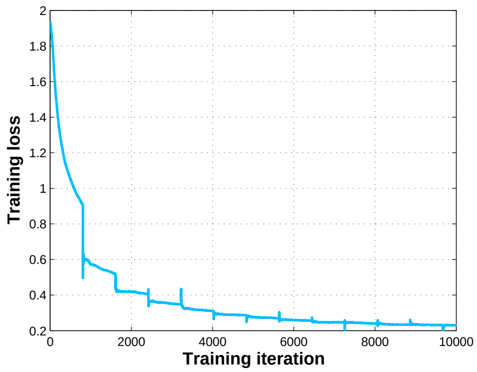


Fig. 12. Training loss with different iteration times.

efficacy of CAM and FAM, we remove the module of CAM and FAM from DeFusionNet at each recurrent step, then the rest of the network (CVPR19) is the same as [40]. We also present the F-measure, MAE and AUC scores of CVPR19 on the three datasets in Table 2. The corresponding PR curves, F-measure curves and ROC curves are plotted in Figure 8, Figure 9 and Figure 10. As can be seen, without CAM and FAM, the final results are obviously degraded, which demonstrates the efficacy of the CAM and FAM. In Figure 13, we give some visual results with/without CAM and FAM. As can be seen, with CAM and FAM, DeFusionNet can focus on the most discriminative features and weaken the influence of noisy features, which produces more pure detected results. In addition, inorder to validate the efficacy of each single component, we only remove CAM or FAM from DeFusionNet, and denote the rest part as "with FAM and without CAM" (wFAMwoCAM) and "with CAM and without FAM" (wCAMwoFAM). The corresponding F-measure, MAE and AUC scores on different datasets are also shown in Table 2.

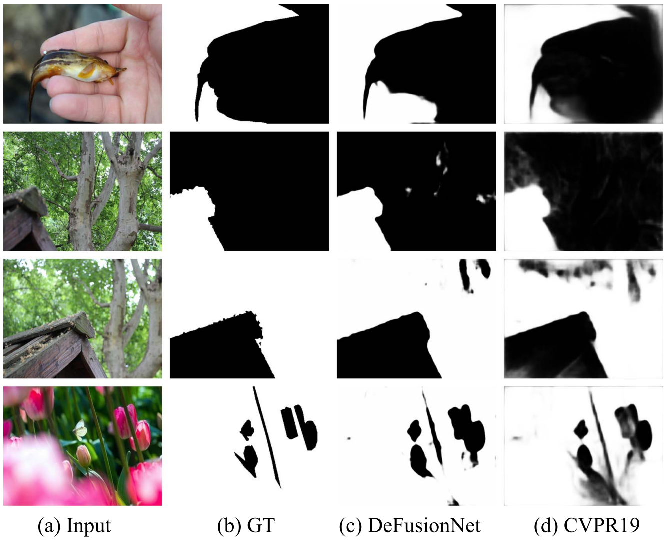


Fig. 13. Visual comparison of detected defocus blur maps generated by DeFusionNet with/without CAM and FAM. The results demonstrate that DeFusionNet can obtain more accurate results by using CAM and FAM.

Effectiveness of GAP and GMP. In fact, both GAP and GMP associate the feature maps with the final output. However, GMP just focuses on the most important region of a feature map while GAP focuses on every region of a feature map. In practical, for human visual system, the in-focus objects in an image attract more attention. On the one hand, GMP can help the network select the most important region which represent in-focus part of an image. In such a manner, GMP determines whether a region is blurry, thus it reflects the defocus intensity. On the other hand, GAP takes all of the regions into consideration, it helps the network to distinguish different blurry regions from an image, even the regions are with different defocus intensity, therefore, it reflects the size of blurry regions. In Figure 14, we give two visual results of DeFusionNet with/without GAP/GMP. As can be seen, our DeFusionNet without GAP (denoted as noGAP) can suppress some noisy regions, but the defocus blur regions are not complete. On the contrary, our DeFusionNet without GMP (denoted as noGMP) can detect the complete blurry regions, but the results are mixed with some noisy regions. By using both GMP and GAP, DeFusionNet can detect more pure defocus blur regions from an image.

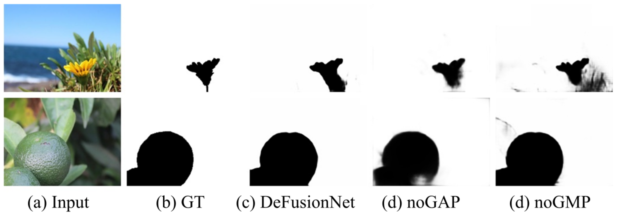


Fig. 14. Visual comparison of detected defocus blur maps generated by DeFusionNet with/without GMP/GAP.

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 and AUC scores in Table 2 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, MAE and AUC scores by using different times of recurrent step in Table 3. In Figure 15, we also give some visual results of different time steps in Figure 15. As can be seen from Table 3 and Figure 15, 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 2
Ablation analysis using F-measure, MAE and AUC scores

Methods	Shi et al.'s dataset			DUT			CTCUG		
	F_β	MAE	AUC	F_β	MAE	AUC	F_β	MAE	AUC
woFFRM	0.909	0.152	0.830	0.904	0.126	0.631	0.876	0.155	0.676
CVPR19	0.917	0.116	0.836	0.922	0.115	0.632	0.891	0.138	0.689
wFAMwoCAM	0.921	0.110	0.840	0.931	0.106	0.637	0.894	0.132	0.695
wCAMwoFAM	0.922	0.105	0.842	0.943	0.096	0.640	0.896	0.122	0.699
DeFusionNet_O1	0.913	0.115	0.828	0.932	0.114	0.631	0.894	0.133	0.688
DeFusionNet_O2	0.916	0.116	0.829	0.927	0.113	0.635	0.895	0.135	0.688
DeFusionNet_O3	0.917	0.116	0.831	0.928	0.114	0.636	0.893	0.132	0.689
DeFusionNet_O4	0.918	0.122	0.833	0.934	0.121	0.638	0.892	0.131	0.687
DeFusionNet_O5	0.917	0.114	0.829	0.933	0.112	0.632	0.896	0.132	0.693
DeFusionNet_O6	0.916	0.110	0.834	0.938	0.110	0.637	0.895	0.130	0.694
DeFusionNet	0.925	0.102	0.844	0.952	0.082	0.643	0.899	0.127	0.705

TABLE 3
Ablation analysis of the times of recurrent steps (Step_k represents using k times of recurrent steps in DeFusionNet).

Recurrent Step	Shi et al.'s dataset			DUT			CTCUG		
	F_β	MAE	AUC	F_β	MAE	AUC	F_β	MAE	AUC
Step_1	0.716	0.235	0.772	0.872	0.202	0.585	0.785	0.224	0.621
Step_2	0.899	0.121	0.820	0.913	0.122	0.627	0.859	0.176	0.676
Step_3	0.925	0.102	0.844	0.952	0.082	0.643	0.899	0.127	0.705
Step_4	0.925	0.101	0.845	0.952	0.081	0.644	0.899	0.125	0.706
Step_5	0.926	0.100	0.845	0.954	0.081	0.645	0.901	0.125	0.706
Step_6	0.926	0.100	0.846	0.954	0.080	0.645	0.901	0.125	0.707

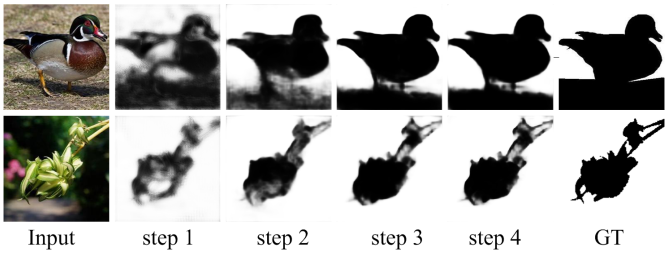


Fig. 15. Visual results at different time steps.

4.5 Challenges of the New Dataset

As introduced in Subsection 4.1, in order to validate defocus blur detection algorithms, we collect a new dataset CTCUG by considering some challenging cases such as complex background, in-focus areas with low contrast, in-focus background and out-of-focus foreground, and same class of objects with different defocus condition. The last four rows of Figure 11 show some results obtained by different defocus blur detection methods. As can be seen, nearly all of the algorithms fail to well separate the defocus blur regions from original input images. For example, in the forth row from the last, some of plant leaves in the input images are in-focus while some of plant leaves are out-of-focus, and all of the plant leaves have the same color and texture, which makes the separation of defocus blur regions difficult. There are some complex background in the input image of the second row from the last, the results of different methods are also affected by the background clutter.

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 semantic features. Finally, DeFusionNet obtains the final accurate defocus blur map by fusing the outputs from all the layers. By considering that different scales of receptive views produce the features with different extents of discrimination, we add a channel attention module after the feature fusing at each step to select more discriminative features to refine the layer-wise features. In order to narrow the gap between different feature extraction layers, we embed a feature adaptation module before In addition, in order to promote further study and evaluation of different defocus blur detection models, we collect a new dataset consists of 150 challenging images and their pixel-wise defocus blur annotations. Extensive experimental results demonstrate that the proposed DeFusionNet consistently outperforms other state-of-the-art methods in terms of both accuracy and efficiency.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments for improving this paper and also like to thank NVIDIA Corporation for the donation of a Titan Xp GPU card and a Titan V GPU card used for this research.

References

[1] S. Alireza 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.

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

[3] F. Couzinie-Devy, J. Sun, K. Alahari, and J. 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.

[4] C. Dong, C. 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.

[5] 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.

[6] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[7] X. Hu, L. Zhu, J. Qin, C.-W. Fu, and P.-A. Heng. Recurrently aggregating deep features for salient object detection. In *AAAI*, pages 6943–6950, 2018.

[8] R. Huang, W. Feng, M. Fan, L. Wan, and J. Sun. Multiscale blur detection by learning discriminative deep features. *Neurocomputing*, 285:154–166, 2018.

[9] 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.

[10] P. Jiang, H. Ling, J. Yu, and J. Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *Proceedings of the IEEE international conference on computer vision*, pages 1976–1983, 2013.

[11] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[12] 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.

[13] B. Kim, H. Son, S.-J. Park, S. Cho, and S. Lee. Defocus and motion blur detection with deep contextual features. *Computer Graphics Forum*, 37(7):277–288, 2018.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[15] J. Lee, S. Lee, S. Cho, and S. Lee. Deep defocus map estimation using domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12222–12230, 2019.

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

[17] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):2999–3007, 2017.

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

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[20] K. Ma, H. Fu, T. Liu, Z. Wang, and D. Tao. Deep blur mapping: Exploiting high-level semantics by deep neural networks. *IEEE Transactions on Image Processing*, 27(10):5155–5166, 2018.

[21] B. Masia, A. Corrales, L. Presa, and D. Gutierrez. Coded apertures for defocus deblurring. In *Symposium Iberoamericano de Computacion Grafica*, volume 5, 2011.

[22] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

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

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

[25] 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 2760–2769, 2017.

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

[27] Y. Qi, S. Zhang, L. Qin, Q. Huang, H. Yao, J. Lim, and M.-H. Yang. Hedging deep features for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

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

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

[30] 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.

[31] A. Shocher, N. Cohen, and M. Irani. "zero-shot" super-resolution using deep internal learning. In *IEEE Conference on computer vision and pattern recognition*, pages 3118–3126, 2018.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Representation Learning*, 2015.

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

[34] C. Sun, H. Lu, and M.-H. Yang. Learning spatial-aware regressions for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8962–8970, 2018.

[35] 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. IEEE, 2009.

[36] C. Tang, C. Hou, Y. Hou, P. Wang, and W. Li. An effective edge-preserving smoothing method for image manipulation. *Digital Signal Processing*, 63:10–24, 2017.

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

[38] C. Tang, P. Wang, C. Zhang, and W. Li. Salient object detection via weighted low rank matrix recovery. *IEEE Signal Processing Letters*, 24(4):490–494, 2017.

[39] 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.

[40] C. Tang, X. Zhu, X. Liu, L. Wang, and Z. Albert. Defusionnet: Defocus blur detection via recurrently fusing and refining multi-scale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2700–2709, 2019.

[41] C. T. Vu, T. D. Phan, and D. 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.

[42] X. Wang, B. Tian, C. Liang, and D. Shi. Blind image quality assessment for measuring image blur. In *Congress on Image and Signal Processing*, pages 467–470. IEEE, 2008.

[43] S. Xie and Z. Tu. Holistically-nested edge detection. In *IEEE international conference on computer vision*, pages 1395–1403, 2015.

[44] G. Xu, Y. Quan, and H. 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.

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

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

[47] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1857–1866, 2018.

[48] K. Zhang, W. Zuo, Y. Chen, D. 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.

[49] S. Zhang, X. Shen, Z. Lin, R. Mech, J. P. Costeira, and J. M. Moura. Learning to understand image blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6586–6595, 2018.

[50] W. Zhang and W.-K. Cham. Single image focus editing. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1947–1954. IEEE, 2009.

[51] W. Zhang and W.-K. Cham. Single-image refocusing and defocusing. *IEEE Transactions on Image Processing*, 21(2):873–882, 2012.

[52] 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.

[53] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, pages 294–310, 2018.

[54] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[55] W. Zhao, X. Hou, X. Yu, Y. He, and H. Lu. Towards weakly-supervised focus region detection via recurrent constraint network. *IEEE Transactions on Image Processing*, 29:1356–1367, 2019.

[56] W. Zhao, F. Zhao, D. Wang, and H. 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.

[57] W. Zhao, F. Zhao, D. Wang, and H. Lu. Defocus blur detection via multi-stream bottom-top-bottom network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[58] W. Zhao, B. Zheng, Q. Lin, and H. Lu. Enhancing diversity of defocus blur detectors via cross-ensemble network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8905–8913, 2019.

[59] 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.

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

Chang Tang (M'16) received his Ph.D. degree from Tianjin University, Tianjin, China in 2016. He joined the AMRL Lab of the University of Wollongong between Sep. 2014 and Sep. 2015. He is now an associate professor at the School of Computer Science, China University of Geosciences, Wuhan, China. Dr. Tang has published 30+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-MM, IEEE T-KDE, IEEE T-HMS, IEEE SPL, ICCV, CVPR, IJCAI, AAAI and ACMM, etc. He served on the Technical Program Committees of IJCAI 2018/2019/2020, ICME 2018/2019/2020, AAAI 2019/2020, CVPR 2019/2020 and ICCV 2019/2020. His current research interests include machine learning and computer vision.

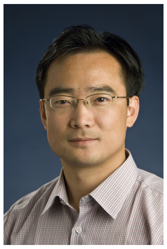
Xinwang Liu (M'13) received his PhD degree from National University of Defense Technology (NUDT), China. He is now Associate Professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, T-KDE, T-IP, T-NNLS, T-MM, T-IFS, NIPS, ICCV, CVPR, AAAI, IJCAI, etc.

Wanqing Li (SM'12) received his PhD in electronic engineering from The University of Western Australia. He was an Associate Professor (91-92) at Zhejiang University, a Senior Researcher and later a Principal Researcher (98-03) at Motorola Research Lab, and a visiting researcher (08,10 and 13) at Microsoft Research US. He is currently an Associate Professor and Director of Advanced Multimedia Research Lab (AMRL) of University of Wollongong, Australia. His research areas include machine learning, 3D computer vision, 3D multimedia signal processing and medical image analysis.

Dr. Li is a Senior Member of IEEE. He serves as an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Multimedia and Journal of Visual Communication and Image Representation.

Xiao Zheng received the master's degree from Tianjin Medical University, Tianjin, P. R. China, in 2014. She is currently pursuing the Ph.D. degree from School of Computer, National University of Defense Technology, Changsha, China. Her recent research interests include computer vision and machine learning.

Jian Xiong (M'13) received the BS degree in engineering, and the MS and PhD degrees in management from the National University of Defense Technology, Changsha, China, in 2005, 2007, and 2012, respectively. He is an associate professor with the School of Business Administration, Southwestern University of Finance and Economics. His research interests include data mining, multi objective evolutionary optimization, multiobjective decision making, project planning, and scheduling. He is a member of the IEEE.



Lizhe Wang received the B.E. and M.E. degrees from Tsinghua University, Beijing, China, and the Dr.Eng. (Magna Cum Laude) degree from the University of Karlsruhe, Karlsruhe, Germany.

He is currently a ChuTian Chair Professor with the School of Computer Science, China University of Geosciences, Beijing, and also a Professor with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing. His research interests include high-performance computing, eScience, and remote sensing image processing. Dr. Wang is a fellow of the IET and the British Computer Society. He serves as an Associate Editor of the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Cloud Computing, and the IEEE Transactions on Sustainable Computing.

Albert Y. Zomaya is currently the Chair Professor of High Performance Computing & Networking in the School of Computer Science, University of Sydney. He is also the Director of the Centre for Distributed and High Performance Computing which was established in late 2009. Professor Zomaya was an Australian Research Council Professorial Fellow during 2010-2014 and held the CISCO Systems Chair Professor of Internetworking during the period 20022007 and also was Head of school for 20062007 in the same school.

Prior to his current appointment he was a Full Professor in the Electrical and Electronic Engineering Department at the University of Western Australia, where he also led the Parallel Computing Research Laboratory during the period 19902002. He served as Associate, Deputy, and ActingHead in the same department, and held numerous visiting positions and has extensive industry involvement.

Professor Zomaya published more than 550 scientific papers and articles and is author, co-author or editor of more than 20 books. He served as the Editor in Chief of the IEEE Transactions on Computers (2011-2014). Currently, Professor Zomaya serves as a Founding Editor in Chief of the IEEE Transactions on Sustainable Computing, Founding Co-Editor in Chief of the IET Cyber-Physical Systems, and Associate Editor-in-Chief (Special Issues), Journal of Parallel and Distributed Computing. He also serves as associate editor for 22 leading journals, such as, the ACM Computing Surveys, ACM Transactions on Internet Technology, and IEEE Transactions on Cloud Computing. Professor Zomaya is the Founding Editor of several book series, such as, the Wiley Book Series on Parallel and Distributed Computing, Springer Scalable Computing and Communications, and the IET Book Series on Big Data. He is a Fellow of the IEEE.

Antonella Longo received the Ph.D. degree in information engineering, in 2004. She teaches data management and big data management for supporting decision making at management engineering and business school master courses. She is currently an Assistant Professor with the Department of Engineering for Innovation, University of Salento. She carries out her research activity at the Software Engineering and Telemedia Laboratory (SET- Lab), University of Salento, where she coordinates the research activities about service modeling and computing, and the applications in smart cities. Her research interests deal with information systems and databases, service-oriented architectures design for cloud infrastructure, technology-enhanced learning, and citizen science. Her current research interests include big data management and exploration of cloud architecture integration with edge computing in smart cities. On these topics, she published more than 80 articles in peer-reviewed journals and international conference proceedings.