Defocus Blur Detection by Fusing Multiscale Deep Features With Conv-LSTM

HONGJUN HENG, HEBIN YE, AND RUI HUANG (Member, IEEE)
College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China
Corresponding author: Rui Huang (rhuang@cauc.edu.cn)

This work was supported in part by the United Funds of the Civil Aviation Administration of China and the National Natural Science Foundation (NSFC) under Grant U1333109, and in part by the Scientific Research Project of Tianjin Education Commission under Grant 2019KJ126.

ABSTRACT Defocus blur detection aiming at distinguishing out-of-focus blur and sharpness has attracted considerable attention in computer vision. The present blur detectors suffer from scale ambiguity, which results in blur boundaries and low accuracy in blur detection. In this paper, we propose a defocus blur detector to address these problems by integrating multiscale deep features with Conv-LSTM. There are two strategies to extract multiscale features. The first one is extracting features from images with different sizes. The second one is extracting features from multiple convolutional layers by a single image. Our method employs both strategies, i.e., extracting multiscale convolutional features from same image with different sizes. The features extracted from different sized images at the corresponding convolutional layers are fused to generate more robust representations. We use Conv-LSTMs to integrate the fused features gradually from top-to-bottom layers, and to generate multiscale blur estimations. The experiments on CUHK and DUT datasets demonstrate that our method is superior to the state-of-the-art blur detectors.

INDEX TERMS Defocus blur detection, multiscale, blur detection, Conv-LSTM.

I. INTRODUCTION
Defocus blur or out-of-focus blur is a common problem when shooting photos. Defocus blur is mainly due to the distance of the object from the focal plane. Some blurs are intentional to highlight the salient objects, while others are unwanted and must be removed. Beside blur removal, defocus blur detection (DBD) is also widely applied in blur magnification [1], [2], image deblurring [3]–[5], saliency object detection [6]–[9], and image quality assessment [10], [11].

Traditional DBD methods use gradient [12]–[16], frequency [17]–[21], SVD [22], LBP [23], and sparse representation [24] to conduct blur detection. Although these hand-crafted features are easily extracted, they lack the distinguishing capability for complex scenes. As shown in Fig. 1 (c), hand-crafted feature-based method, DBDF [18], fails to detect the blurred background marked with the yellow rectangle. With the development of deep learning (DL), an increasing number of blur detectors [25]–[31] are designed on the basis of convolutional neural networks (CNNs). DL-based methods learn effective features for distinguishing blurred and sharp parts from labeled images without well-designed hand-crafted features. In these methods, semantic and low-level features are integrated to generate robust blur detection. Compared with traditional blur detectors, DL-based blur detectors, DeFusionNet [27] and BTBCRL [28], achieve effective blur detection results (Figs. 1 (d) and (e)). However, DeFusionNet and
BTBCRL fail to distinguish blurred and sharp features in homogeneous and low-contrast regions.

Apart from homogeneity and low-contrast, scale ambiguity is an import problem in blur detection. As shown in Fig. 2, the vegetable leaves are sharp in Scale 1 and Scale 2 while remaining blurred in Scale 3. A defocus blur detector that integrates multiscale deep features with Conv-LSTM [32] is proposed in this paper to generate robust blur detection. We adopt VGG16 [33] as the basic feature extractor to extract multiscale deep convolutional features from images with different sizes to solve the scale problem and to generate robust blur detection. Specifically, we first extract multiscale convolutional features from images with different sizes. Then the features extracted from different sized images at the corresponding convolutional layer are fused to generate more robust representation. These fused features are integrated by Conv-LSTMs from top-to-bottom layers to estimate multiscale blur maps. With this manner, the proposed method can fully explore the semantic features of top layers and low-level spatial features of bottom layers. Thus, our method can effectively alleviate misclassification in homogeneous and low-contrast regions, and robust to clutter background. We use multiple losses as supervisors on blur map of each scale blur maps. With this manner, the proposed method can fully explore the semantic features of top layers and low-level spatial features of bottom layers. Thus, our method can effectively alleviate misclassification in homogeneous and low-contrast regions, and robust to clutter background. We use multiple losses as supervisors on blur map of each Conv-LSTM to improve accuracy. The loss function is the weighted summation of cross-entropy loss, Precision, Recall, F-measure, and MAE. We compare our method with 9 state-of-the-art blur detectors on CUHK [18] and DUT [28] datasets. The experimental results demonstrate that the proposed method is superior to the other methods. The main contributions are three-fold.

- Our method uses two types of scales, that is, image size and feature scale, to solve the scale ambiguity of blur detection and achieves superior blur detection results.
- We propose to use Conv-LSTM to integrate high-level semantic information and low-level spatial information for pursuing accurate blur detection in homogeneous and low-contrast regions.
- We use multiple losses, e.g., cross-entropy loss, Precision, Recall, F-measure and MAE as our loss functions to help our network learn effective features.

II. RELATED WORK

A. HAND-CRAFTED FEATURE-BASED METHODS

Traditional DBD methods are based on hand-crafted features, such as, frequency, gradient, or other low-level cues, in which blurred image patches have smoother gradients, smaller singular values and fewer high-frequency components than the sharp ones [17], [25]. Thus, the gradient strength or SVD value can be used as the measurement of the blur level. Golestaneh and Karam [17] proposed a method for blur detection based on a high-frequency multiscale fusion and sort transform, which uses DCT coefficients of the gradient magnitudes from multiple resolutions. Shi et al. [18] deliberately developed local blur features, including local filters, gradient distribution, and spectral measure, to enhance discriminative power for blur detection. Zhang and Hirakawa [19] proposed a double discrete wavelet transform (DDWT) that is designed to sparsify the blurred image and the blur kernel simultaneously. Tang et al. [21] used a log-averaged spectrum as a blur metric for image blur region detection. Xiao et al. [22] proposed a blur metric based on the fusion of multiscale single value features in gradient domains and inferred the blur maps in multiple bands. Yi and Eramian [23] introduced an image sharpness metric based on local binary patterns (LBPs) to separate in- and out-of-focus image regions. Shi et al. [24] developed a blur feature via sparse representation and image decomposition to conduct blur detection, which directly establishes the correspondence between sparse edge representation and blur strength estimation.

The traditional methods need well-designed features and fine-tuned classifiers for blur detection, which hinders general research use. Furthermore, traditional blur detectors obtain poor blur detection results when images containing homogeneous or low-contrast regions.

B. DEEP LEARNING-BASED METHODS

DL has been recently successfully applied in saliency detection [6]–[9], image super-resolution [35] and recognition [33], and other computer vision problems. Previous trials using DL for blur detection showed promising performance [25]–[28]. Park et al. [25] combined hand-crafted features with deep blur features at image patch-level and fed the combined features into a fully connected neural network classifier to determine the amount of defocus blur. Zeng et al. [26] trained CNN with super-pixel and extracted effective filters by PCA. The blurred maps are produced by convolving the image with the selected filters. In [28], Zhao et al. proposed a multi-stream bottom-top-bottom fully convolutional network and cascaded residue learning network to solve the DBD problems. Tang et al. [27] recommended a deep convolutional network to fuse the features from shallow and deep layers for blur detection. The feature fusion and refinement were performed step-by-step in recurrent manner.

Although deep blur detectors achieve superior blur detection performance to traditional blur detectors, blur detection results can still be improved by introducing new and effective
modules for blur detectors. We introduce Conv-LSTM [32] in our defocus blur detector and achieve promising blur detection results.

III. METHOD

Fig. 3 shows the network architecture of the proposed defocus blur detector. We partition it into multiscale feature extraction sub-network (MsFEN) and multiscale blur estimation sub-network (MsBEN). MsFEN has three branches extracting multiscale convolutional features from three resized images. Then the features of the same level convolutional layers from different branches are concatenated together and fused by convolutional layers to generate more robust representations. Thus, the proposed network is robust to blur scale ambiguity by considering information from two types of scale, image scale and feature scale. MsBEN is built by Conv-LSTMs and takes the fused features as input to estimate multiscale blur maps. We adopt top-to-bottom strategy to integrate the features of lower layers by Conv-LSTMs for progressively refine the estimated blur maps. In following section, we will introduce the model architecture, multi-layer losses and implementation details of our method.

A. MODEL ARCHITECTURE

1) MULTISCALE FEATURE EXTRACTION SUB-NETWORK

VGG16 [33] is a popular network used in blur detection [27], [28]. We also use VGG16 as our basic feature extractor. To make it is suitable for our blur detection task, we remove the last pooling layer and last two fully connected layers to have the same size with the estimated blur maps. In following section, we will introduce the model architecture, multi-layer losses and implementation details of our method.

Let I denote an input image. We first resize I into three different scales, denoted as I1, I2, and I3 with sizes of 320 × 320, 256 × 256 and 192 × 192, respectively. We use MsFEN to extract multiscale convolutional features Fconv from I, where s = 1, 2, 3 denotes image scale and l = 1, . . . , 5 denotes layer index. Owing to the different dimensions of Fconv1, Fconv2, and Fconv3, we use bilinear interpolation to resize Fconv2 and Fconv3 to have the same size with Fconv1 for feature fusion. The resized feature can be formulated by

\[ F_{\text{ups}} = \Phi_{\text{ups}}(\text{Conv}(F_{\text{conv}}, 1 \times 1 \times 64), H^l, W^l), \]

where \( H^l \) and \( W^l \) are the height and width, respectively, of the feature maps of the l-th layer of I; \( \Phi_{\text{ups}} \) denotes the bilinear interpolation upsampling; Conv(·, 1 × 1 × 64) denotes convolutional layers with filters of 1 × 1 × 64. We use this kind of convolutional layer before upsampling to change the channel numbers of the convolutional features to 64. We also apply batch normalization (BN) and ReLU following the convolutional layers. ReLU are used as default after convolution without special description.

The obtained features of the same level convolutional layers from different branches are fused by convolutional layers. Specifically, we first concatenate \( F_{\text{conv}}^1 \), \( F_{\text{ups}}^1 \), and \( F_{\text{ups}}^3 \) and then fuse them with a convolutional layer to obtain fused feature \( F_{\text{conv}1} \), which can be formulated by

\[ F_{\text{conv}1} = \text{Conv} (\text{Cat}(F_{\text{conv}1}, F_{\text{ups}}^2, F_{\text{ups}}^3), 3 \times 3 \times 64), \]

where \( \text{Cat}(\cdot) \) denotes concatenate operation. By concatenating the convolutional features, we can obtain more robust features to conquer the blur scale ambiguity.

2) MULTISCALE BLUR ESTIMATION SUB-NETWORK

With the fused features \( F_{\text{conv}1} \), we conduct top-to-bottom integration by feeding them to Conv-LSTMs. The estimated blur map \( B_5 \) produced by \( F_{\text{conv}5} \) is computed by

\[ B_5 = \Psi_{\text{Conv-LSTM}} (F_{\text{conv}5}), \]

where \( \Psi_{\text{Conv-LSTM}}(\cdot) \) denotes computation of the Conv-LSTM, which will be introduced in the following section.

To gradually refine the blur maps, we concatenate the upsampled blur map at the i + 1 layer with the fused features of \( F_{\text{conv}i} \), because features of the bottom layer contain more fine structure information. Then the concatenated features are input into Conv-LSTM to estimate blur map \( B_i \). The mathematical formulation can be denoted as

\[ B_i = \Psi_{\text{Conv-LSTM}} (\text{Cat}(\sigma(F_{\text{conv}i}), \Phi_{\text{ups}}(B_{i+1}, H^l, W^l))), \]

where \( \sigma(\cdot) \) is a sigmoid activation function. We use sigmoid function to normalize the values of \( F_{\text{conv}i} \) into [0, 1] because

| Conv block | Layers | Kernel size | Scale 1 | Feature size Scale 2 | Scale 3 |
|------------|--------|-------------|---------|----------------------|---------|
| 1          | Conv2D Conv2D# MaxPooling | 64×3×3 64×3×3 2×2 | 320×320 256×256 192×192 |
| 2          | Conv2D Conv2D# MaxPooling | 128×3×3 128×3×3 2×2 | 160×160 128×128 98×96 |
| 3          | Conv2D Conv2D# MaxPooling | 256×3×3 256×3×3 2×2 | 80×80 64×64 48×48 |
| 4          | Conv2D Conv2D# MaxPooling | 512×3×3 512×3×3 2×2 | 40×40 32×32 24×24 |
| 5          | Conv2D Conv2D# MaxPooling | 512×3×3 512×3×3 2×2 | 20×20 16×16 12×12 |
H. Heng et al.: DBD by Fusing Multiscale Deep Features With Conv-LSTM

FIGURE 3. The network architecture of the proposed defocus blur detector. Please see text for details.

$B_{t+1} \in [0, 1]$. With Conv-LSTM layers, we gradually integrate the blur information of the top layers with features of bottom layers to robustly generate accurate blur detection result. Fig. 4 shows the visual comparison of the blur maps estimated by different Conv-LSTM layers. We can find that the blur maps are refined progressively.

3) CONV-LSTM
Conv-LSTM is developed from the traditional fully connected LSTM [34] for capturing spatially-related features. This layer uses convolutional operations instead of dot products to code spatial information. Fig. 5 shows the detailed architecture of Conv-LSTM used in this paper.

Similar to LSTM, Conv-LSTM has three gates, namely, input gate $i_t$, output gate $o_t$, and forget gate $f_t$, to control the information flow. Cell status $C_t$ and hidden state $H_t$ are updated by the values of $i_t$, $o_t$ and $f_t$. Let $X_t$ denote the input of Conv-LSTM layer at each time step. If $i_t$ is activated, then the input will be accumulated to the cell $C_t$. If $f_t$ is activated, then the past cell status $C_{t-1}$ will be neglected. The information propagation of $C_t$ to $H_t$ is controlled by the output gate $o_t$. The update process at time step $t$ can be expressed as follows:

$$i_t = \sigma(W^i_t \ast X_t + W^H_t \ast H_{t-1} + b_i)$$  \hspace{1cm} (5)
$$f_t = \sigma(W^f_t \ast X_t + W^H_t \ast H_{t-1} + b_f)$$  \hspace{1cm} (6)
$$o_t = \sigma(W^o_t \ast X_t + W^H_t \ast H_{t-1} + b_o)$$  \hspace{1cm} (7)
$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W^c_t \ast X_t + W^H_t \ast H_{t-1} + b_c)$$  \hspace{1cm} (8)
$$H_t = o_t \odot \tanh(C_t)$$  \hspace{1cm} (9)

FIGURE 4. Blur maps estimated by Conv-LSTM5-1 are shown in (a)-(e), respectively.
where $\sigma$ and $tanh$ are activation functions of sigmoid and hyperbolic tangent, respectively; $\ast$ is the convolution operation; $\circ$ is the Hadamard product; $W$ and $b$ are the learned weights and biases, respectively. The max time step is set to 3 in our experiments. We use a convolutional layer with the filter of $1 \times 1 \times 1$ on the hidden state of last time step to get the estimated blur map.

**B. MULTI-LAYER LOSSES**

In recent DL-based methods, multiple losses [37] are used as supervisors for fast convergence. We supervise the blur map at each scale. Specifically, we resize the blur maps to $320 \times 320$ same as input size, and these blur maps are used to compute the grounds with truth. Different from image segmentation that only uses cross-entropy loss function, we utilize Precision, Recall, F-measure, and MAE as part of the loss function to obtain results of high Precision, Recall, and F-measure, while low MAE and cross-entropy. Thus, the loss used in this paper is defined as follows:

$$L_{Blur}(B, G) = L_C(B, G) + \alpha_1 L_{F3}(B, G) + \alpha_2 L_p(B, G) + \alpha_3 L_R(B, G) + \alpha_4 L_{MAE}(B, G)$$

(10)

where $B$ is blur map, $G$ is groundtruth. We set $\alpha_1$, $\alpha_2$, $\alpha_3$, $\alpha_4$ to 0.1 by experience.

**Cross-entropy loss $L_C$:**

$$L_C(B, G) = \frac{1}{W \times H} \sum_p [G_p \cdot \log(B_p + \epsilon) + (1 - G_p) \cdot \log(1 - B_p + \epsilon)]$$

(11)

where $B_p$ and $G_p$ are the pixel values at position $p$ of blur map and groundtruth; $W$ and $H$ are the width and height of input image, respectively; and $\epsilon$ is a small constant to avoid zero division error.

**Precision loss $L_p$:**

$$L_p(B, G) = -\frac{\sum_p (B_p \cdot G_p)}{\sum_p B_p + \epsilon}$$

(12)

Recall loss $L_R$:

$$L_R(B, G) = -\frac{\sum_p (B_p \cdot G_p)}{\sum_p G_p + \epsilon}.$$  

(13)

F-measure loss $L_{F3}$:

$$L_{F3}(B, G) = \frac{(1 + \beta^2) \cdot L_p(B, G) \cdot L_R(B, G)}{\beta^2 \cdot L_p(B, G) + L_R(B, G) + \epsilon}.$$  

(14)

where $\beta^2 = 0.3$ as suggested by [36]. Negative values are used to minimize the loss because the high F-measure, Precision, and Recall values indicate the improved performance of the network.

**MAE loss $L_{MAE}$:**

$$L_{MAE}(B, G) = \frac{1}{W \times H} \sum_p |B_p - G_p|.$$  

(15)

MAE is the mean absolute error, which reflects the similarity between the output and the groundtruth.

The total loss of our network is defined as follows:

$$\mathcal{L} = \sum_{l=1}^{5} L_{Blur}(B_l, G),$$  

(16)

where $B_l$ is the blur map produced by $l$-th Conv-LSTM layer.

**C. IMPLEMENTATION DETAILS**

1) DATA AUGMENTATION

We first pre-train our network on synthesized blur data and fine-tune it on blur benchmark datasets.

$a$: SYNTHESIZED BLUR DATA

We randomly pick 2000 images from the Berkeley segmentation dataset (BSDS) [39], uncompressed color image dataset (UCID) [40] and PASCAL 2008 dataset [41]. We then use the method proposed by [28] to synthesize the out-of-focus blurred images. Specifically, we utilize Gaussian blur with kernel size $7 \times 7$ and $\sigma = 2$ to blur the top-, bottom-, left-, and right-half image regions. We use the same blur kernel to blur the image regions five times to simulate different blur extents. Thus, each image can generate 20 blurred images. We obtain
a total of 40,000 synthetic blur images for pre-training. Fig. 6 shows some examples of synthesized blur images and the corresponding groundtruth.

![synthesized blur images and groundtruth](image)

**FIGURE 6.** Examples of synthesized defocus blur images and their corresponding groundtruths. The black and white regions in GT denotes blur and sharp, respectively.

**b: REAL BLUR DATA**

We employ flipping, rotation, and cropping to augment real blur images. Specifically, we rotate images with 11 random angles with zero padding and flip the image in horizontal, vertical and horizontal-vertical directions. We crop the images with aspect ratios beyond [0.95, 1.05] into two images. Finally, the number of the training images is approximately 16 times to the number of the original images.

2) TRAINING DETAILS

The model is implemented by Keras on Titan Xp GPU. We use the weights of VGG16 trained on ImageNet as the initialization of our feature extraction networks. The newly added layers are initialized via “Xavier Uniform”. For pre-training, we set the learning rate to $5 \times 10^{-7}$ and batch size to 2. We stop pre-training after three epochs because the network is converged after three epochs. For fine-tuning, we set the learning rate to $5 \times 10^{-8}$ and decreased it by $1 \times 10^{-5}$ every five epochs until $1 \times 10^{-8}$. The bath size is set to 2. We stop training when the network is converged. The optimizer is Adam [38].

IV. EXPERIMENT

A. DATASET

In the experiments, we use two publicly available defocus blur detection datasets, DUT and CUHK, for evaluating the proposed method.

The DUT dataset [28] contains 600 and 500 images for training and testing, respectively. The images on the CUHK dataset have complex scenes and cluster background. For a fair comparison, we utilize two testing and training set partition strategies. The first partition is CUHK1, which is the same as Tang et al. [27]. The second partition is CUHK2 which is proposed by Zhao et al. [28].

B. EVALUATION CRITERIA

We use F-measure and MAE to evaluate different blur detection methods. F-measure is a comprehensive weighted indicator using precision and recall. A high value denotes improved performance. Our F-measure computation is same with [28], [29]. Specifically, we resize the blur map to $320 \times 320$ and obtain the binary map by OTSU. We compute the precision and recall of the binary map. Finally, the F-measure can be computed as follows:

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}},$$

where $\beta^2 = 1$ and $\beta^2 = 0.3$ are used in our experiments like other blur detection methods [27], [29], [30].

The MAE is computed by the blur map and groundtruth without binary, which can be computed as

$$\text{MAE} = \frac{1}{W \times H} \sum_p |B_p - G_p|$$

where $W$ and $H$ are width and height of the image, respectively. $B_p$ and $G_p$ are values of blur map and groundtruth at position $p$, respectively.

C. COMPARISON OF METHODS

We utilize nine state-of-the-art blur detection methods for comparison. DHDE [25], BTBCRL [28], and DeFusionNet [27] are DL-based blur detectors. JNB [24], DBDF [18], HiFST [17], LBP [23], SVD [15], and SS [21] are hand-crafted feature-based blur detectors. We use the results provided by the authors or by Tang et al. [27].

D. EXPERIMENTAL RESULTS ANALYSIS

1) QUALITATIVE ANALYSIS

Fig. 7 shows the blur detection results of different blur detectors on images with different scenes. Scene1 contains images with simple scenes, but has large homogeneous region. Scene2 contains images with cluttered background. Scene3 contains images that have similar foreground and background colors.

Scene1 shows that SS [21] and DeFusionNet [27] failed in detecting the large homogeneous regions in the first and third images, respectively. DHDE [25] cannot suppress the values of the blurred backgrounds in all images. Although HiFST [17] generates high values on the sharp regions, its boundaries are blurred. Compared with other blur detectors, the blur detection results of our proposed method are accurate in homogeneous regions.
Scene 2 shows three images with cluster backgrounds. The first and third images have bright spots in the blurred backgrounds. Almost all compared blur detectors treat the bright spots as sharp regions. However, our proposed method accurately detect sharp regions. In the second image, the compared methods fail to detect the doll’s body, while our method detected the integral sharp body region.

Scene 3 shows the blur detection results of two images sharing similar foregrounds and backgrounds. BTBCRL [28] fails to detect the real sharp regions from the blurred backgrounds. HiFST [17] treats the blurred background as sharp. SS [21] and DHDE [25] misclassify the blurred background into a sharp region. Compared with other blur detectors, our method achieves correct blur detection results on two representative images.

The above analysis indicates that, the proposed blur detection method is robust in different scenarios and superior to state-of-the-art blur detection methods.

2) QUANTITATIVE ANALYSIS
We report $F_{0.3}$, $F_1$, and MAE of different blur detectors on DUT, CUHK1, and CUHK2 in Table 2. Except for
TABLE 2. Quantitative comparison of $F_{0.3}$, $F_1$ and MAE scores of different blur detectors. The first, second and third ranked results are highlighted with red, green and blue, respectively.

| Dataset | Metric | JNB | DBDF | HiFST | LBP | SVD | SS | DIIDE | BTBCRL | DeFusionNET | Ours |
|---------|--------|-----|------|-------|-----|-----|-----|-------|---------|-------------|------|
| DUT     | $F_{0.3}$ | 0.568 | 0.625 | 0.714 | 0.754 | 0.683 | 0.737 | 0.663 | 0.828 | 0.831 | 0.873 |
|         | $F_1$    | 0.539 | 0.592 | 0.696 | 0.750 | 0.666 | 0.730 | 0.627 | 0.806 | 0.815 | 0.874 |
|         | MAE      | 0.419 | 0.378 | 0.306 | 0.195 | 0.299 | 0.243 | 0.401 | 0.139 | 0.119 | 0.074 |
|         | $F_{0.3}$ | 0.699 | 0.690 | 0.789 | 0.775 | 0.731 | 0.765 | 0.738 | –     | 0.865 | 0.848 |
| CUHK1   | $F_1$    | 0.693 | 0.660 | 0.786 | 0.770 | 0.699 | 0.747 | 0.713 | –     | 0.864 | 0.847 |
|         | MAE      | 0.355 | 0.343 | 0.238 | 0.211 | 0.301 | 0.266 | 0.390 | –     | 0.118 | 0.129 |
|         | $F_{0.3}$ | 0.827 | 0.718 | 0.833 | 0.851 | 0.841 | 0.806 | –     | 0.919 | –     | 0.932 |
|         | MAE      | 0.395 | 0.291 | 0.225 | 0.154 | 0.236 | 0.224 | –     | 0.087 | –     | 0.053 |

FIGURE 8. Comparison of PR curves, Precision, Recall and F-measure ($\beta^2 = 0.3$) bars of different blur detectors on DUT ((a) and (b)) and CUHK2 ((c) and (d)) datasets.

DHDE [25], the DL-based blur detectors are better than hand-crafted feature-based methods. Our method achieves the highest $F$-measure values and the lowest MAE values on the DUT and CUHK2 datasets. On the CUHK1, our method ranks second place, while DeFusionNET [27] ranks first.

On the DUT dataset, $F_{0.3}$, $F_1$, and MAE of the proposed method are 0.873, 0.874 and 0.074, respectively. Compared with the state-of-the-art blur detectors, our method is the only one whose $F_{0.3}$ and $F_1$ are higher than 0.87 and MAE is lower than 0.1. Compared with DeFusionNET [27], our method achieves 5.1% and 7.2% relative $F_{0.3}$ and $F_1$ improvements, respectively, and 37.8% relative to MAE reduction.

Except for BTBCRL [28], DHDE [25] and DeFusionNET [27] conducted their experiments on the CUHK1 dataset. On this dataset, DeFusionNET shows the best blur detection results. The third-ranked method is HiFST [17]. Compared with HiFST, our method achieves 7.5% and 7.8% relative to $F_{0.3}$ and $F_1$ improvements, respectively, and 45.8% relative to MAE reduction.

We conduct our experiment on CUHK2 for comparison with BTBCRL [28]. The results of DHDE [25] and DeFusionNET [27] are not reported on this dataset. On CUHK2, compared with BTBCRL, our method achieves 1.4% and 3.0% relative to $F_{0.3}$ and $F_1$ improvements, respectively, and 39.1% relative to MAE reduction.

We also report the PR curves, Precision, Recall and F-measure bars of different blur detection methods on two benchmark datasets in Fig. 8. We can find that our Precision, Recall and F-measure values are the highest among the compared methods on both datasets. Note that the PR curves of our method are flat when Recall smaller than 0.8, because our results are close to the binary blur map whose Precision and Recall are not sensitive to the threshold.

From the above analysis, the proposed method is superior to other state-of-the-art blur detection methods.

3) RUNNING TIME

We report the running time of different blur detection methods for processing an image with size of $320 \times 320$ in Table 3. Except BTBCRL [28] and DeFusionNET [27] use the running time reported in their papers, the running time of all other compared methods are conducted on same computer. Our method is a bit slower than DeFusionNET. Note that it is tested on Titan Xp. Our method takes 0.06s for processing an image, which is faster than other compared methods.

TABLE 3. Comparison of running time of different blur detection methods to process an image with size of $320 \times 320$.

| Method | JNB | DBDF | HiFST | LBP |
|--------|-----|------|-------|-----|
| Time (s) | 3.7 | 97.5 | 82.0 | 9.9 |
| Method | SVD | DeFusionNET | BTBCRL | Ours |
| Time (s) | 19.5 | 0.056 | 12 | 0.06 |

E. ABLATION STUDIES

1) USING DIFFERENT BASIC FEATURE EXTRACTORS

We replace VGG16 with VGG13 and VGG19, and report their performance on DUT dataset in Table 4. From the table, we can find that using deeper networks can improve the performance of defocus blur detection. However, deeper network
uses more computation resources. We find that VGG16 has similar performance with VGG19 but takes less running time. Thus, we use VGG16 as our basic feature extractor.

2) EFFECTIVENESS OF MULTISCALE FEATURE
The proposed network utilizes three image scales with input image sizes of $320 \times 320$, $256 \times 256$ and $192 \times 192$. We compare three networks with different combinations of the input image sizes to verify the effectiveness of the proposed multiscale feature extraction. The first network uses all three input image sizes and is dubbed Net-S$^3$. The second network adopts two input image sizes of $320 \times 320$, $256 \times 256$, which is called Net-S$^2$. The third network, Net-S$^1$, only uses input image sizes of $320 \times 320$.

We use $F_0$, $F_1$, and $F_2$ to evaluate the performance of Net-S$^1$, Net-S$^2$, and Net-S$^3$ on the DUT and CUHK datasets, respectively. The results are shown in Table 5. Results show that all criteria of Net-S$^3$ are better than those of Net-S$^1$ and Net-S$^2$. On the DUT dataset, Net-S$^1$ and Net-S$^2$ have similar performances. This similarity demonstrates that using only two scales as inputs cannot promote the blur detection performance on the DUT dataset. Meanwhile, Net-S$^3$ is better than Net-S$^1$ and Net-S$^2$ on the CUHK dataset. Net-S$^2$ shows better blur detection performance than that of Net-S$^1$ with improvements. Although $F_0$, $F_1$ and $F_2$ of Net-S$^3$ are slightly better than Net-S$^2$, it is also better than Net-S$^1$ and Net-S$^2$ both.

TABLE 5. Comparison of $F_0$, $F_1$ and $F_2$ of the proposed method with different combinations of image scales on DUT and CUHK$^2$ datasets. The best results are highlighted in bold.

| Method   | DUT       | CUHK$^2$   |
|----------|-----------|------------|
|          | $F_0$     | $F_1$      | $F_2$      | $F_0$     | $F_1$      | $F_2$      |
| Net-S$^3$| 0.873     | 0.874      | 0.074      | 0.932     | 0.932      | 0.053      |
| Net-S$^2$| 0.869     | 0.869      | 0.077      | 0.931     | 0.930      | 0.053      |
| Net-S$^1$| 0.870     | 0.869      | 0.077      | 0.927     | 0.927      | 0.056      |

The above analysis shows that the multiscale feature extraction network effectively extracted multiscale deep features, and successfully achieved feature fusion between different scales.

3) EFFECTIVENESS OF CONV-LSTM
We replace the Conv-LSTM layers with convolutional layers in blur estimation sub-network to verify the effectiveness of Conv-LSTM. The first variant is replacing Conv-LSTM with a convolutional layer, named Single-Conv. The second one is replacing Conv-LSTM with three convolutional layers, which has the same parameter number with the corresponding Conv-LSTM, denoted as Multi-Conv. The details of these layers are shown in Table 6. We retrain these networks with the same training data and report $F_0$, $F_1$, and $F_2$ of the DUT and CUHK$^2$ datasets in Table 7. We can find that all evaluation criteria using Conv-LSTM layers are better than those using convolutional layers.

TABLE 6. Comparison of $F_0$, $F_1$ and $F_2$ of the proposed network by using Conv-LSTM, Single-Conv and Multi-Conv on DUT and CUHK$^2$ datasets. The best results are highlighted in bold.

| Method      | DUT       | CUHK$^2$   |
|-------------|-----------|------------|
| Conv-LSTM   | 0.873     | 0.874      | 0.074      | 0.932     | 0.932      | 0.053      |
| Single-Conv | 0.871     | 0.872      | 0.077      | 0.925     | 0.927      | 0.056      |
| Multi-Conv  | 0.871     | 0.871      | 0.076      | 0.928     | 0.931      | 0.053      |

V. CONCLUSION
We propose a defocus blur detector in this paper to solve the blur scale ambiguity. Our method employs two strategies to obtain scale robust blur detection results, that is, different image sizes and multiscale convolutional features. We recommend the use of Conv-LSTM layers to integrate the multiscale features. Thus, high-level semantic and low-level spatial features are effectively incorporated together to generate accurate blur detection. We utilize the weighted loss of cross-entropy, F-measure, Precision, Recall and MAE to supervise the network. The network is pre-trained with synthesized blur images and fine-tuned on real blur images. The experimental results on two benchmark datasets show that our method is superior to the other state-of-the-art blur detection methods.

REFERENCES
[1] S. Bae and F. Durand, “Defocus magnification,” Comput. Graph. Forum, vol. 26, no. 3, pp. 571–579, Sep. 2007.
[2] F. Kriener, T. Binder, and M. Wille, “Accelerating defocus blur magnification,” Proc. SPIE, vol. 8667, Mar. 2013, Art. no. 86671Q.
[3] R. Huang, M. Fan, Y. Xing, and Y. Zou, “Image blur classification and unintentional blur removal,” IEEE Access, vol. 7, pp. 106327–106335, 2019.
[4] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, “Blind image deblurring using dark channel prior,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1628–1636.
[5] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, “Scale-recurrent network for deep image deblurring,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8174–8182.
[6] P. Jiang, H. Ling, J. Yu, and J. Peng, “Salient region detection by UFO: Uniqueness, focussness and objectness,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 1976–1983.

[7] W. Wang, J. Shen, X. Dong, and A. Borji, “Salient object detection driven by fixation prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1711–1720.

[8] R. Huang, W. Feng, Z. Wang, Y. Xing, and Y. Zou, “Exemplar-based image saliency and co-saliency detection,” Neurocomputing, vol. 371, pp. 147–157, Jan. 2020.

[9] R. Huang, Y. Xing, and Z. Wang, “RGB-D salient object detection by a CNN with multiple layers fusion,” IEEE Signal Process. Lett., vol. 26, no. 4, pp. 552–556, Apr. 2019.

[10] L. Li, W. Lin, X. Wang, G. Yang, K. Bahrami, and A. C. Kot, “No-reference image blur assessment based on discrete orthogonal moments,” IEEE Trans. Cybern., vol. 46, no. 1, pp. 39–50, Jan. 2016.

[11] X. Wang, B. Tian, C. Liang, and D. Shi, “Blind image quality assessment for measuring image blur,” in Proc. Congr. Image Signal Process., May 2008, pp. 467–470.

[12] E. Saad and K. Hirakawa, “Defocus blur-invariant scale-space feature extractions,” IEEE Trans. Image Process., vol. 25, no. 7, pp. 3141–3156, Jul. 2016.

[13] S. Zhao and T. Sim, “Defocus map estimation from a single image,” Pattern Recognit., vol. 44, no. 9, pp. 1852–1858, Sep. 2011.

[14] R. Liu, Z. Li, and J. Jia, “Image partial blur detection and classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 24–26.

[15] B. Su, S. Lu, and C. L. Tan, “Blurred image region detection and classification,” in Proc. IEEE/ACM Conf. Multimedia, 2011, pp. 1397–1400.

[16] Y. Pang, H. Zhu, X. Li, and X. Li, “Classifying discriminative features for blur detection,” IEEE Trans. Cybern., vol. 46, no. 10, pp. 2220–2227, Oct. 2016.

[17] S. A. Golestaneh and L. J. Karam, “Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 596–605.

[18] J. Shi, L. Xu, and J. Jia, “Discriminative blur detection features,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 2965–2972.

[19] Y. Zhang and K. Hirakawa, “Blur processing using double discrete wavelet transform,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 1091–1098.

[20] C. T. Yu, T. D. Phan, and D. M. Chandler, “Sj: A spectral and spatial measure of local perceived sharpness in natural images,” IEEE Trans. Image Process., vol. 21, no. 3, pp. 934–945, Mar. 2012.

[21] L. 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 Process. Lett., vol. 23, no. 11, pp. 1652–1656, Nov. 2016.

[22] H. Xiao, W. Lu, R. Li, N. Zhong, Y. Yeung, J. Chen, F. Xue, and W. Sun, “Defocus blur detection based on multiscale SVD fusion in gradient domain,” J. Vis. Commun. Image Represent., vol. 59, pp. 52–61, Feb. 2019.

[23] X. Yi and M. Eramian, “LBP-based segmentation of defocus blur,” IEEE Trans. Image Process., vol. 25, no. 4, pp. 1626–1638, Apr. 2016.

[24] J. Shi, L. Xu, and J. Jia, “Just noticeable defocus blur detection and estimation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 657–665.

[25] J. Park, Y.-W. Tai, D. Cho, and I. S. Kweon, “A unified approach of multi-scale deep and handcrafted features for defocus detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1746–1755.

[26] X. Zeng, Y. Wang, J. Mao, J. Liu, W. Peng, and N. Chen, “A local metric for defocus blur detection based on CNN feature learning,” IEEE Trans. Image Process., vol. 28, no. 5, pp. 2107–2115, May 2019.

[27] C. Tang, X. Zhu, X. Liu, L. Wang, and A. Zomaya, “DeFusionNET: Defocus blur detection via recurrently fusing and refining multi-scale deep features,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2700–2709.

[28] W. Tang, F. Zhao, D. Wang, and H. Lu, “Defocus blur detection via multi-stream Bottom-Top-Bottom network,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Mar. 25, 2019, doi: 10.1109/TPAMI.2019.2906588.

[29] X. Wang, S. Zhang, X. Liang, H. Zhou, J. Zheng, and M. Sun, “Accurate and fast blur detection using a pyramid M-Shaped deep neural network,” IEEE Access, vol. 7, pp. 86611–86624, 2019.

[30] R. Huang, W. Feng, M. Fan, L. Wan, and J. Sun, “Multiscale blur detection by learning discriminative deep features,” Neurocomputing, vol. 285, pp. 134–166, Apr. 2018.

[31] W. Zhao, F. Zhao, D. Wang, and H. Lu, “Defocus blur detection via multi-stream Bottom-Top-Bottom fully convolutional network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3080–3088.

[32] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 802–810.

[33] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[34] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[35] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, Feb. 2016.

[36] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, “Frequency-tuned salient region detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 1597–1604.

[37] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, “Deeply-supervised nets,” in Proc. Artif. Intell. Statist., 2015, pp. 562–570.

[38] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[39] P. Arbelaz, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 5, pp. 898–916, May 2011.

[40] G. Schaefer and M. Stich, “UCID: An uncompressed color image database,” Proc. SPIE, vol. 5307, pp. 472–480, Dec. 2003.

[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, The PASCAL Visual Object Classes Challenge 2008 (VOC 2008) Results. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html

HONGJUN HENG received the Ph.D. degree from Nankai University under the supervision of Prof. Z. Wang. He is currently an Associate Professor with the Department of Computer Science and Technology, Civil Aviation University of China. His research interests include intelligent information processing and computer application, specifically include natural language processing and knowledge graph.

HEBIN YE received the B.S. degree from the Hebei University of Engineering, Handan, China. He is currently pursuing the M.S. degree with the Civil Aviation University of China. His research interest includes computer vision, especially blur detection.

RUI HUANG (Member, IEEE) received the Ph.D. degree from Tianjin University under the supervisions of Prof. W. Feng and J. Sun. He is currently a Lecturer with the Department of Computer Science and Technology, Civil Aviation University of China. His research interests include computer vision, machine learning, and visual surveillance, specifically include saliency and co-saliency detection, image blur detection, and generic pattern recognition.