Rethinking the Defocus Blur Detection Problem and A Real-Time Deep DBD Model

Ning Zhang[1,2[0000−0001−8128−7573]], Junchi Yan[1,2[0000−0001−9639−7679]⋆]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
ningzh6610@gmail.com, yanjunchi@sjtu.edu.cn

Abstract. Defocus blur detection (DBD) is a classical low level vision task. It has recently attracted attention focusing on designing complex convolutional neural networks (CNN) which make full use of both low level features and high level semantic information. The heavy networks used in these methods lead to low processing speed, resulting difficulty in applying to real-time applications. In this work, we propose novel perspectives on the DBD problem and design convenient approach to build a real-time cost-effective DBD model. First, we observe that the semantic information does not always relate to and sometimes mislead the blur detection. We start from the essential characteristics of the DBD problem and propose a data augmentation method accordingly to inhibit the semantic information and enforce the model to learn image blur related features rather than the semantic features. A novel self-supervision training objective is proposed to enhance the model training consistency and stability. Second, by rethinking the relationship between defocus blur detection and salience detection, we identify two previously ignored but common scenarios, based on which we design a hard mining strategy to enhance the DBD model.By using the proposed techniques, our model that uses a slightly modified U-Net as backbone, improves the processing speed by more than 3 times and performs competitively against state of the art methods. Ablation study is also conducted to verify the effectiveness of each part of our proposed methods.

Keywords: Defocus blur detection, self-supervision, hard-mining

1 Introduction

Deep learning techniques have promoted explosive growth of many computer vision tasks including but not restricted to image classification [17], object location and detection [14], semantic segmentation [1], salience detection [5]. However, the performance of above algorithms are related to the quality of images and blur images with lots of noise can lead to a sharp decline in accuracy. Thus blur

⋆ Corresponding author is Junchi Yan. Work was partly supported by National Key Research and Development Program of China 2018AAA0100704, NSFC (61972250, U19B2035), and SJTU Global Strategic Partnership Fund (2020 SJTU-CORNELL).

Fig. 1. Examples of blurry images from [16] (zoom in for details). The left half shows the motion blur with a mask (black area denotes blur). This paper focuses on the right half case: defocus blur detection. As highlighted around the red bounding box, yellow leaves show similar semantics, while both in-focus and defocused regions appear.

detection is an fundamental yet challenging topic in computer vision area. It detects the degraded area with loss of image details, which is the basic and critical pre-process step for deblurring. As shown in Fig. 1, image blur can be generally classified into two types: motion blur and out of focus blur (or defocus blur) [16]. Object moving definitely accounts for motion blur. The defocus blur is caused by the limited depth of fields of camera lens. Objects located in too remote or too close distance are out of focus and blurry. The defocus blur is ubiquitous for pictures captured by digital cameras, especially by cell phone cameras. In this paper, we pay attention to the detection of defocus blur areas. Our work is motivated by two important insights which have been ignored and probably misused by previous works. 1) the semantic information does not always relate to and sometimes mislead the blur detection. 2) the salience detection, which is a related topic with DBD, can be used to help the mining of hard occasions in DBD. Based on these two important insights, we designed clever and convenient approaches to solve the DBD problem.

Defocus blur detection methods can generally be divided into two categories: traditional algorithms based on handcrafted features [10–12, 15, 16, 18, 20, 22, 26, 27, 30, 31] and deep learning algorithms based on CNN [6, 21, 25, 28, 29]. The former ones usually apply statistics of gradient or high frequency information to differentiate blur, considering that blur areas are usually smoother than clear areas. They can work for simple cases effectively and endure lousy results for complex scenes. The phenomena are related to the following reasons.

At first, although smoothing or filtering images can unavoidably lead to image blur, we could not judge the clarity just by gradients. Some objects such as sky and ground are always with low gradients all the time, no matter they are focused or not. Besides, some areas with complex texture and high frequency could be out of focus. There are lots of leaves marked in a red rectangle in the second row of Fig. 1. These leaves have similar textures and gradient information, however parts of leaves are out of focused and parts are not. Thus it is unreasonable and not the case for practical application to distinguish focus area just using gradient distributions or frequency information.

Secondly, CNN based methods are more flexible to combine multi-scale of information and multi-level of features together to detect blurry. Multi branches and shortcut fuel information flow and information fusion in deep networks. CNN

can fuse the results of different filters and extract discriminative features. These characteristics of CNN lead to more powerful feature extracting than traditional algorithms based on handcrafted features.

The authors in [28] propose a multi-stream bottom-top-bottom fully convolutional network (BTBnet) to detect defocus blur. They design a recurrent reconstruction strategy which fuses low level cues and high level information to improve the performance. Although the BTBnet achieves impressive results, their large computation cost hinders their wide applications. In [25], a dilated fully convolutional neural network is applied to widen the network without increasing the parameters. A deep defocus blur detector cross ensemble network (CEnet) is proposed in [29]. Two groups of defocus blur detectors are alternatively optimized to enhance diversity in CEnet. While in [21], a deep neural network is devised which recurrently fuses and refines multi-scale deep features (DeFusionnet) for defocus blur detection.

Most of the existing works try to improve the accuracy of the DBD models by designing deeper or wider networks, while other works tried to decrease the computational cost and increase the processing speed of those models. These approaches emphasize too much on the structure of the network and employ the semantic information to detect the defocus blur region. However, we have an insight that the semantic information in the images does not always relate to the blur region. We can observe from Fig. 1 that the regions of similar semantic (yellow leaves) can be either blur or clear. This results in unsatisfactory performance of those approaches. In this paper, we rethink the DBD problem, explore its characteristics and its relationship with related topics. To enforce the neural network to learn the defocus blur related features, rather than the semantic features, we propose a novel data augmentation method by taking advantage of the transition-invariant property of the defocus blur region. A self-supervision objective is proposed to enhance the robustness of the model. In addition, we have another insight that the salient detection can help the mining of the hard occasions in the DBD. It is difficult to detect the in-focus region in the non-salient region, and to detect the defocus region in the salient region, especially when a single object contains both the in-focus part and defocus part. Based on this insight, we proposed a novel hard mining policy to train the neural network. With the proposed method, a simple U-Net [13] with slight modification performed competitively and even better than most of the state of the arts. In addition, our method achieves notable improvement on the processing speed.

Conclusively, the contributions of this paper are as follows.

- We put forward two basic observations, which have been relatively ignored in previous research. First, the semantic information does not always relate to, and sometimes even can mislead the DBD. Existing deep learning based methods are mostly devoted to designing wider and/or deeper networks to learn the semantic features. While little study has been made to consider the inherent mechanism of the DBD problem. Second, we take a closer look at the connection between the salience detection and DBD, and find that the salient information can help to locate the hard occasions in DBD.

- Based on the first observation, we propose a novel data augmentation technique to inhibit the semantic information and enforce the neural network to learn the blur related features rather than semantic features. A self-supervision objective is devised to enhance the consistency of training. To our best knowledge, this is the first work to incorporate the blur related constraints into the objective for deep learning.
- Based on the second observation, we design a hard mining approach to cope with two hard occasions, which are previously overlooked while practically common: 1) defocus and salient region; 2) in-focus but non-salient region. These two scenarios are more difficult to identify from other scenarios. We propose a hard mining strategy for these two cases. Our resulting method can handle these two cases effectively without sacrificing the performance on other scenarios. It is empirically shown that the hard mining policy can improve the performance notably, especially on the regions of depth boundary.
- With the proposed objective, even by using a simple network, i.e. a slightly modified U-Net, we can achieve competitive and even superior performances, with improved processing speed by more than 3 times. Experimental results show that the proposed method performs competitively. We also conducted an ablation study to verify the effectiveness of self-supervision objective and our hard mining technique.

2 Related Works

In general, defocus blur detectors can be divided into two categories: traditional methods using hand-designed features and deep learning methods.

2.1 Traditional Methods

The blurry images are relatively smoother in some scenes. Inspired from that, some researchers made full use of gradient information or frequency information to detect blur. The works [2,19] detect the DBD using the radio of strong gradient components in an image patch. The authors in [11] design special kernels to measure image sharpness. In [18], singular value distribution and gradient distribution work together for blur detection. Multi-scale high frequency information and sorted transform of coefficients of gradient magnitudes are fused to detect blur in [3]. Fourier domain features are applied to detect image sharpness and a public blur detection dataset has been built in [16]. The authors in [20] obtain coarse-to-fine blurred region using spectral and spatial information. In [23], image patch ranks are fully used to estimate blur map.

Although hand designed features have made contributions to detect blur regions, they often fail in complex scenes. Compared with traditional algorithms, CNN methods can fuse multi-scale multi-level information to different blurred and clear image regions and outperform most of hand designed methods.

2.2 Deep Network based Methods

In [12], hand designed features and deep learning features are used together to estimate blur region. However, these deep learning features are extracted in local patches and time consuming. Subsequently, more carefully-designed CNN structures are proposed for DBD. The authors in [28] propose a multi-stream bottom-top-bottom fully convolutional networks to estimate the probability of each pixel being out-of-focus and blurry. It is a fusion and recurrent reconstruction network which is deep and wide. It integrates both low-level and high-level information to handle blur images. [25] apply a dilated fully convolutional neural network which increases the field-of-view without increasing parameters. The cross-ensemble network is designed to obtain multiple defocus blur detectors with less computation cost [29]. In [21], a novel network which fuses deep features and suppressed background clutter is also devised.

3 Proposed Method

3.1 Approach Overview

It is sometimes assumed that CNN can make full use of semantic information and this information benefit detection performance [21, 28]. While this assumption may not always hold. Image regions with similar semantic information can be easily broken into in-focused and out-of-focus parts, which is a common occurrence for images captured by macro lens. As shown in the right half in Fig. 1, there is a clear boundary between the focused paper glass and defocus leaves. In this respect, semantic information is beneficial to distinguish sharp object from the blur. However the leaves in red rectangle are blurry due to out of focus and other leaves are clear. All of them are considered as leaves and can not be distinguished according to semantic information. It means that there is no necessary relation between the semantic information and the defocus detection. The semantic information sometimes disturb the judgment of defocus or not. We should pay attention to the image clarity itself. In addition, we identify two hard occasions for the DBD problem: 1) detection of the defocus blur region at the salient region, 2) detection of the in-focus region at the non-salient region. Based on the two insights, we propose 1) a novel data augmentation method to inhibit the semantic information, and a self-supervision objective to enhance the model consistency. The proposed strategy not only expands the training but also reduces the affect of semantic information. 2) a hard mining strategy by taking advantage of the relation between the salience detection and DBD.

We expound our algorithm in the following three parts. First, we explain the data augmentation strategy and the self-supervision loss. Then the hard mining strategy is introduced by analyzing the relation of blur detection and the salience detection. Finally, we present our network which is a slightly modified U-Net.

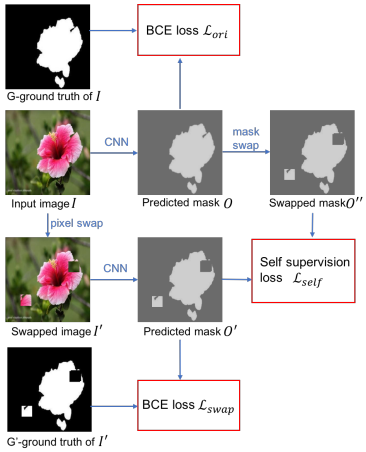

Fig. 2. Self-supervision learning scheme. For input image I, we obtain the swapped input I' by swapping the pixel values of two patches. The predicted mask of swapped input image O' should be same with O'' which is the swapped mask of original image.

3.2 Data Augmentation and Self-supervision Loss

As our goal is to detect the blur region, rather than the blur objects, it is more important to learn the image clarity features than the semantic features. For a patch in a given image, whether it is blur or not is irrelevant to its location in the image, neither relevant to what object it belongs to. A clear patch remains to be clear wherever we move it in an image. Similarly, a blur patch is still blur even we paste it to a clear region.

Take Fig. 2 as an example, it gets the focus on the flower, and all the background areas are out of focus and blurry. If we move the position of flower, it would lead to some artifacts and sharpness on the boundary. However, this action did not cause the flower to be blurry or the backgrounds to be clear. If we moved parts of flower, it would break the integrity of flower and inhibit the semantic information. However, this action still would not change the clarity of the flower parts.

Formally speaking, for an input image I, we randomly choose two patches with the same size and swapped the pixel value of one patches for the other. Then we got the swapped image I', as shown in Fig. 2. G and G' are corresponding ground truth of the image I and the swapped image I', respectively. O and O' represent the predicted masks for the image I and the swapped image I'.

Most existing DBD works train models to minimize the distance between the groudtruth G and the predicted blurry mask O. The distance is computed as the binary cross entropy (BCE) between G and O. This can be expressed as:

$$\mathcal{L}_{ori} = \sum_{i=1}^{h}\sum_{j=1}^{w} -G(x,y) * \log(O(x,y)) \tag{1}$$

where, h and w stand for the height and width of images respectively. While (x, y) represents the coordinates of images.

In this paper, we enlarge the training set tremendously by swapping the image patches. In this way, we should also train the model on the augmented data to minimize the distance between the swapped ground truth G' and the swapped predicted mask O'. This is represented as Eq. (2).

$$\mathcal{L}_{aug} = \sum_{i=1}^{h} \sum_{j=1}^{w} -G'(x, y) * \log(O'(x, y)) \tag{2}$$

In addition, according to our analysis, the predicted mask O' should be a derivative of predicted mask O. For robustness, the transition of input images should lead to the same transition of the output mask. This can be stated as:

$$O = \Phi_\theta(I); \quad f(O) = \Phi_\theta(f(I)) \tag{3}$$

where Φ_θ is a neural network of DBD parametered by θ. $f(I) = I'$ and $f(O) = O''$ are the patch transition operation. In our case, the swapped output O'' and the output of the swapped image O' should be same for a robust DBD system. Inspired by this, we introduced the self-supervision loss to enhance the robustness of our model. The similarity of the output O'' and the output O' are calculated by L1 loss.

$$\mathcal{L}_{self} = \frac{1}{h * w} \sum_{i=1}^{h} \sum_{j=1}^{w} |O'(x, y)) - O''(x, y)| \tag{4}$$

The total loss for the proposed self-supervision method is the weighted sum of the above losses.

$$\mathcal{L}_{ts} = \mathcal{L}_{ori} + \mathcal{L}_{aug} + \lambda_s \mathcal{L}_{self} \tag{5}$$

where λ_s is the weight for the self-supervision term. As in our training settings, the number of input original image I and the number of input augmented image I' are always equal, we simply keep the weights of $\mathcal{L}_{ori}$ and $\mathcal{L}_{aug}$ to be 1. And we only adjust the weight λ_s for $\mathcal{L}_{self}$. The patch size used in this paper is 64×64.

3.3 Hard Mining

It is hard to identify the blur of the boundary area, when the clear and the blurred area mingle, especially when a single object contains both focused part and defocus part. DBD algorithms do not have satisfactory performances in above two cases [28] [29]. In this paper, we propose a hard mining algorithm to improve the performances.

By analyzing the DBD related topics, we observe that the salient detection is closely related to but inconsistent with the DBD problem. Salient detection can help to locate the regions of the above difficult scenarios.

Specifically, salience detection is to identify the most visually distinctive regions in images. Fig. 3 shows the difference between the focused area and salient

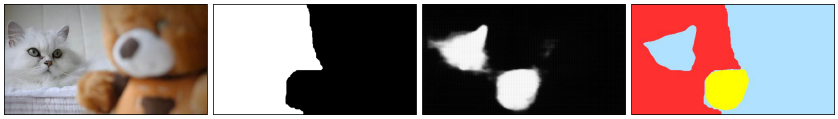

Fig. 3. The relation between the focus detection and salience detection. From the first column to the last column are: the input images (collected in [28]), the ground truth of DBD (black area denotes blur), the salience detection result of [5] (black area denotes non-salient), and the difference between the focused detection and salience detection (yellow area denotes 'salient but out-of-focus', and red area denotes 'in-focus but not salient', blue area denotes 'salient and in-focus' and 'non-salient and out-of-focus').

area. The first column shows the image with out-of-focus area collected by [28]. The second column is the in-focus area of this image. Both the cat and its background are in-focus. The third column contains the salience detection result of one of state of the art [5]. The face of the cat and the paw of the bear is detected as salient. The last column shows the difference of the focus area and the salient area. This difference help us to locate the depth boundary and ignore the semantic information.

According to the above analysis, we creatively proposed a region based hard mining method for DBD. At first, we calculate the salient detection areas by the algorithms in [5], which is a state of the art salient detection method. Secondly, we compute the hard mining region H as 1) the out-of-focus and salient region 2) in-focus but not salient region. Formally, $H = (S \cup F) - (S \cap F)$, where H denotes the hard mining region, S and F denote the salient region and in-focus region, respectively.

By giving different weights to the intersection and the union, we realize a hard mining algorithm based on pixel position. In this way, the loss functions in Eq. (1) and Eq. (2) should be changed into Eq. (6) and Eq. (7), respectively.

$$\mathcal{L}_{weighted} = \sum_{i=1}^{h}\sum_{j=1}^{w} -W(x,y) * G(x,y) * \log(O(x,y)) \tag{6}$$

$$\mathcal{L}_{weightedaug} = \sum_{i=1}^{h}\sum_{j=1}^{w} -W'(x,y) * G'(x,y) * \log(O'(x,y)) \tag{7}$$

where $W(x,y)$ represents the weights based on intersection and difference. $W'(x,y)$ means the swapped weights according to pixel value swap of input images.

In this way, the final loss function of proposed algorithm is defined as follows.

$$\mathcal{L}_{total} = \mathcal{L}_{weighted} + \mathcal{L}_{weightedaug} + \lambda_s \mathcal{L}_{self} \tag{8}$$

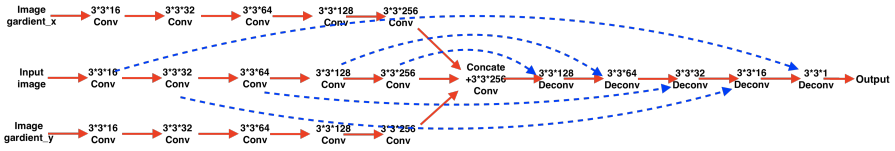

Fig. 4. The backbone network used in this paper. This is a modified U-Net to facilitate the flow of the horizontal and vertical gradient information.

3.4 Backbone Network

We briefly introduce the network used in this paper. As shown in Fig. 4, it is inspired by the design of U-Net [13]. The "$3*3*t$ Conv" module consists of two convolutional filers, two Relu [9] layers and two BN [7] layers and one pooling layer. The convolution kernel size of this module is 3×3 and the number of filters is t. The strides of all the "$3*3*t$ Conv" modules are 2 except for the one before concatenation. Many traditional algorithms apply gradient information for DBD. Inspired that, partial derivatives in both X and Y direction are encoded in our network. Both encoded partial derivatives and the image are concatenated at the end of the encoding part. "$3*3*256$" module reduces the number of redundant filters and fuses features. After that, a dropout layer [4] is added and the dropout rate is set to be 0.2.

The dotted arrows in Fig. 4 represent a shortcut from a encoder layer to a decoder layer. "$3*3*t$ Deconv" means that the kernel size for each deconvolution layer [24] is 3×3 and the number of filters is t. The stride of each deconvolution is 2. Since our proposed algorithm is focused on loss design instead of network design, we just apply this popular network for experiments.

4 Experiments

4.1 Experimental Setup

We use two public and popular defocus blur detection datasets for evaluation.

The CUHK dataset [16] consists of 1000 blurry images. Among these, 296 blur images are caused by object motion and 704 images are defocus blur images. We divide the 704 images into training set and testing set, as many researchers do [21, 28, 29]. There are 604 images randomly selected for training, and the rest are applied to measure the performance in [28, 29]. [21] applied top 604 images for training and the last 100 images for testing. To compare with state of the arts, we share the same training set and testing set with [28, 29].

The DUT dataset appears in [28]. It contains 1100 images with pixel-wise annotations. 600 images belongs to the training set and 500 images belongs to the testing set. It is relatively more challenging dataset.

The proposed method are implemented by PyTorch, and are performed using 4 TitanXP GPUs. The Adam [8] is selected as optimizer. The momentum is set to be 0.9 and weight decay is set to be $5e^{-4}$. The start learning rate is $2e^{-5}$ and decreases to $2e^{-6}$ after 2000 epochs. The batch size is set to 64. It takes about two days to train the network. The swapped patch size is set to be 16. All the weights of hard mining regions are set to be double weights of other areas.

The training images are resized into 320×320 at first step. After that, we apply horizon mirroring and rotating to augment data. Different from other works [21, 28], no extra data is used to pretrain our model.

Evaluation Metrics Three metrics are applied to evaluate the performance of proposed method: F-measure, mean absolute error (MAE), F-measure curve by all the thresholds and the Precision-Recall (PR) curve. The output need to be binarized to calculated F-measure and MAE. F-measure is defined by:

$$F_\beta = \frac{(1+\beta^2) * Precision * Recall}{\beta^2 * Precision + Recall} \tag{9}$$

where $\beta^2 = 0.3$ is employed to emphasize the importance of precision, which represents for the percentage of correctly detected pixels which are focused. The correctly detected pixels is divided by the ground truth number of focused pixels to get Recall. Bigger F-measure means better performance.

The metric MAE denote the average pixel-wise difference between the output results and the ground truth. It is defined as the follow.

$$MAE = \frac{1}{h * w} \sum_{i=1}^{h} \sum_{j=1}^{w} |O(x,y) - G(x,y)| \tag{10}$$

where, O stands for the binarized output and G stands for the ground truth. Smaller MAE represented smaller difference between output and ground truth.

Both F-measure and MAE evaluate the performance of different methods by binarized the outputs at the specific threshold. The F-measure curve and the Precision-Recall curve provide more comprehensive displays of performances. The output are binarized at each point in range of [0, 255] to display the F-measure curve and Precision-Recall curve.

4.2 Comparison with Peer Methods

Compared with traditional methods, deep learning methods achieve better performance for DBD [21, 28, 29]. In this way, we just compare our method with these state-of-the-art algorithms. BTBnet is devised in [28]. BTBnet handles input images with different scales and combines all these information to calculate the possibility of blur. It is complex network and high time costing. After that, the work [29] proposes a deep cross ensemble network (CEnet) which reduces the time cost comparable with BTBnet. As the name implies, CEnet makes full use of diversity of networks to produce accurate results. The authors in [21] present

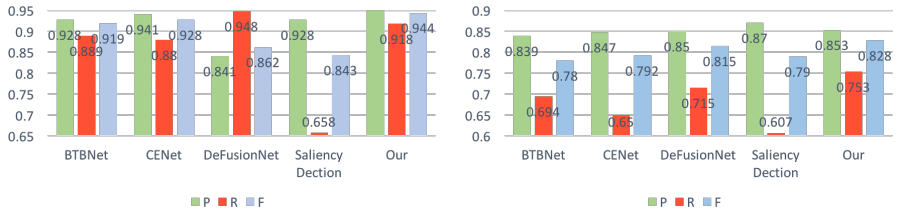

Fig. 5. F-measure of different algorithms. The left plot shows the results over the CUHK dataset. The right plot shows the results over the DUT dataset. P and R stand for precision and recall, respectively. F stands for the F-measure.

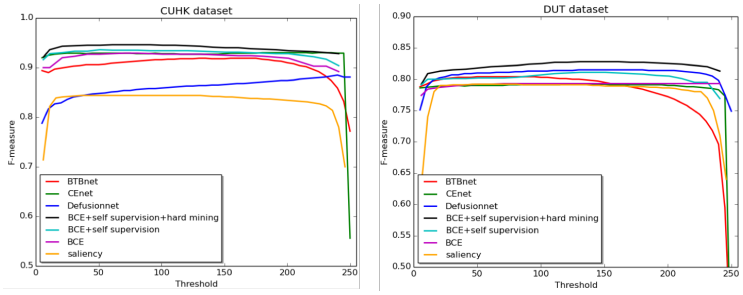

Fig. 6. F-measure by different thresholds of different algorithms. The left plot shows the result over the CUHK dataset. The right plot shows the result over DUT dataset.

Table 1. Comparison of F-measure (the higher the better) and MAE (the lower the better) of different approaches. Two networks (CENet, DeFusionNet) have not released their models hence the exact sizes are unknown and not reported. Best in bold.

Dataset	Metric	Models							
		BTBNet [28]	CENet [29]	DeFusionNet [21]	Salience Detection [5]	BCE+ self-supervision +hard mining	BCE + self-supervision	BCE	BCE without image gradient
CUHK	F-meature↑	0.919	0.928	0.862	0.843	**0.944**	0.933	0.927	0.918
	MAE↓	0.060	0.057	0.111	0.125	**0.053**	0.060	0.066	0.078
DUT	F-meature↑	0.780	0.792	0.815	0.791	**0.828**	0.811	0.793	0.754
	MAE↓	0.127	0.136	0.118	0.151	**0.115**	0.127	0.140	0.163
DUT & CUHK	FPS↑	0.04	15.63	17.86	22.20	**58.82**	**58.82**	**58.82**	**112.35**
	Model Size	200M	-	-	238M	26.73M	26.73M	26.73M	**7.79M**

the DeFusionnet which recurrently fuses and refines multi-scale deep features to detect out-of-focus regions. We just download their results from their project website since they have not released their implementation. As mentioned above, our testing dataset of CUHK dataset is the same with [28, 29] and different from [21]. All the above approaches adopt complex networks and their models are pretrained using ImageNet data [9].

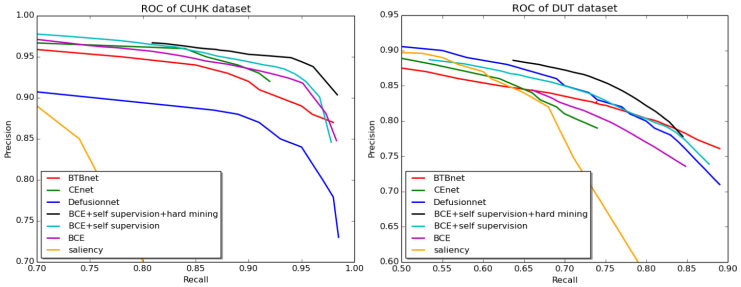

Fig. 7. Precision-Recall curves of different algorithms. The left and right plot shows the result over the CUHK and DUT dataset, respectively.

F-measure and MAE are shown in Table 1 and Fig. 5. Our proposed model (BCE+self supervision+hard mining) outperforms the second best method by 1.6% for F-measure and 2.5% for MAE over the DUT dataset. The testing data of CUHK dataset for our algorithm, BTBnet and CEnet are the same. Our model exceeds CEnet by 1.7% for F-measure and 7.0% for MAE, respectively. The number of the training set and the testing set used by our model and DeFusionnet are the same. However, the images in these set for our model and DeFusionnet are different. Our model beats DeFusionNet by 9.5% for F-measure and 52.3% for MAE. For reference, we also present the salience detection performance on the DBD task, although the salience detection is a different task with the DBD. The salience detection performance is poor on the DBD task over all the metrics.

The F-measure curve and the Precision-Recall curves are shown in and Fig. 6 and Fig. 7, respectively. Our model shows superior performance over most parts of the F-measure curves. Besides, our model generates better results compared with other models over most parts of the Precision-Recall curves.

Fig. 8 and Fig. 9 show some comparison examples of different algorithms. Images in Fig. 8 come from the testing set of the DUT dataset and images in Fig. 9 come from the testing set of the CUHK dataset. We can see that the hard mining method help us to identify the boundary of the focus and defocus regions despite whether they belonging to a same object or not. All the algorithms can detect the main parts of focused area. While the competitors can not get accurate boundary in areas where the focus and defocus changed in high frequency. What's worse, the competitors usually fail to detect the focused part of one object which consists of the focused area and defocus area at the same time.

In the testing phase, each input image is resized into 320×320 pixels to obtain the final defocus blur map. We and [21] both use a single Nvidia GTX Titan Xp GPU for inference, while [28] and [29] use a GTX1080Ti GPU. These two GPUs have similar processing power. As shown in Table 1, our model is highly efficient with the speed of 58.82 FPS (frames per second), which is 3.29 times faster than the second fastest method named Defusionnet.

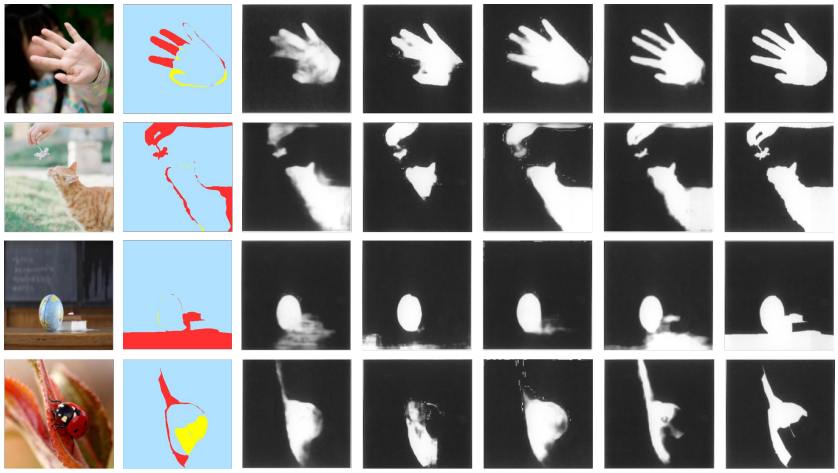

Fig. 8. Examples on DUT (by column): input images, hard mining regions, outputs of BTBnet, CEnet, DeFusionnet, and proposed methods, respectively, and ground truth.

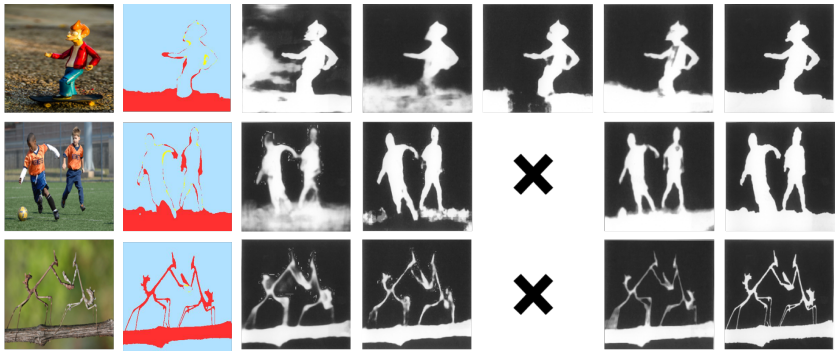

Fig. 9. Examples on CUHK (left to right): input images, hard mining regions, outputs of BTBnet, outputs of CEnet, outputs of DeFusionnet (the bottom two are not public available), outputs of proposed methods, and ground truth.

4.3 Ablation Analysis

We conduct ablation study to test the effectiveness of the proposed self-supervision objective and the hard mining method. The completed proposed method is denoted as 'BCE+self-supervision+hard mining'. While 'BCE+self-supervision' denotes the comparison method which we remove the hard mining part from the whole proposed method. 'BCE' means that we only use the common BCE objective to train our backbone network. 'BCE without image gradient' denotes that the image input gradient information is removed from the backbone network, and only the U-net trained by the common BCE objective is used.

Table 2. Parameter sensitivity test for the self-supervision term weight λ_s.

Dataset	Metric	λ_s					
		0.0	0.1	0.2	0.3	0.4	0.5
CUHK	F-meature↑	0.927	0.932	**0.933**	0.929	0.932	0.928
	MAE↓	0.066	0.063	**0.060**	**0.060**	0.065	0.065
DUT	F-measure↑	0.793	0.807	**0.811**	0.807	0.799	0.805
	MAE↓	0.140	0.131	0.127	**0.124**	0.132	0.131

From Table 1 and Figs. 6, 7, we can observe that the completed proposed method 'BCE+self-supervision+hard mining' significantly outperforms the compared method 'BCE+self-supervision' in all the measurement, while 'BCE+self-supervision' steadily achieves better results than the 'BCE'. These results verify that both the proposed self-supervision objective and the hard mining method are effective for defocus blur detection. In addition, we also show that the if the image gradient is not used, the performance drops obviously, though the model size is much smaller and the process speed is much higher.

4.4 Parameter Sensitive Test

In the above experiments, the weight coefficient λ_s in Eq 5 is set to 0.2. To test the sensitive of the value for λ_s, we test the model 'BCE+self supervision' performance when λ_s is with a set of different values. The testing results are shown in Table 2. We can observe that the performance varies a little as the value of λ_s changes. When $\lambda_s = 0.0$, this model boils down to BCE model, and when $\lambda_s = 0.2$, it is equal to the model $BCE + self\ supervision$ in Table 1.

5 Conclusion

Different from most existing works focusing on designing complex network, in this paper we propose a light model to cope with the DBD problem. We reanalyze the DBD problem and identify that the semantic information may harm the blur detection. Starting from the DBD problem itself, we propose a novel data augmentation method to inhibit the semantic information and enforce the neural network to learn the blur related features rather than the semantic features. A novel self-supervision objective is used to enhance the training. In addition, by analyzing the relation between the salience detection and defocus blur detection, we identify two hard occasions for DBD models, and based on the difference of salience detection and focus detection, we design a hard mining method and give different weights to various parts in image. With the proposed objective function, a simple and slightly modified U-Net can achieve competitive and even better results than competitors whose network are complex, carefully designed and also pretrained on the ImageNet dataset. In addition, our method achieves more than 3 times improvement on processing speed.

References

1. Chen, L., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. arXiv: Computer Vision and Pattern Recognition (2018)
2. Elder, J.H., Zucker, S.W.: Local scale control for edge detection and blur estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(7), 699–716 (1998)
3. Golestaneh, S.A., Karam, L.J.: Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. computer vision and pattern recognition pp. 596–605 (2017)
4. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv: Neural and Evolutionary Computing (2012)
5. Hou, Q., Cheng, M., Hu, X., Borji, A., Tu, Z., Torr, P.H.S.: Deeply supervised salient object detection with short connections. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(4), 815–828 (2019)
6. Huang, R., Feng, W., Fan, M., Wan, L., Sun, J.: Multiscale blur detection by learning discriminative deep features. Neurocomputing **285**, 154–166 (2018)
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv: Learning (2015)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv: Learning (2014)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. neural information processing systems **141**(5), 1097–1105 (2012)
10. Liu, R., Li, Z., Jia, J.: Image partial blur detection and classification. computer vision and pattern recognition pp. 1–8 (2008)
11. Pang, Y., Zhu, H., Li, X., Li, X.: Classifying discriminative features for blur detection. IEEE Transactions on Systems, Man, and Cybernetics **46**(10), 2220–2227 (2016)
12. Park, J., Tai, Y.W., Cho, D., Kweon, I.S.: A unified approach of multi-scale deep and hand-crafted features for defocus estimation. computer vision and pattern recognition (2017)
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. medical image computing and computer assisted intervention pp. 234–241 (2015)
14. Rutishauser, U., Walther, D., Koch, C., Perona, P.: Is bottom-up attention useful for object recognition? computer vision and pattern recognition **2**, 37–44 (2004)
15. Saad, E., Hirakawa, K.: Defocus blur-invariant scale-space feature extractions. IEEE Transactions on Image Processing **25**(7), 3141–3156 (2016)
16. Shi, J., Li, X., Jia, J.: Discriminative blur detection features. In: Computer Vision and Pattern Recognition (2014)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv: Computer Vision and Pattern Recognition (2014)
18. Su, B., Lu, S., Tan, C.L.: Blurred image region detection and classification. ACM multimedia pp. 1397–1400 (2011)
19. Tai, Y., Brown, M.S.: Single image defocus map estimation using local contrast prior. international conference on image processing pp. 1777–1780 (2009)

20. Tang, C., Wu, J., Hou, Y., Wang, P., Li, W.: A spectral and spatial approach of coarse-to-fine blurred image region detection. IEEE Signal Processing Letters **23**(11), 1652–1656 (2016)
21. Tang, C., Zhu, X., Liu, X., Wang, L., Zomaya, A.: Defusionnet: Defocus blur detection via recurrently fusing and refining multi-scale deep features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2700–2709 (2019)
22. Vu, C.T., Phan, T.D., Chandler, D.M.: S3 : A spectral and spatial measure of local perceived sharpness in natural images. IEEE Transactions on Image Processing **21**(3), 934–945 (2012)
23. Xu, G., Quan, Y., Ji, H.: Estimating defocus blur via rank of local patches. Computer Vision and Pattern Recognition pp. 5381–5389 (2017)
24. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. computer vision and pattern recognition (2010)
25. Zhang, S., Shen, X., Lin, Z., Mech, R., Costeira, J.P., Moura, J.M.F.: Learning to understand image blur. Computer Vision and Pattern Recognition pp. 6586–6595 (2018)
26. Zhang, Y., Hirakawa, K.: Blur processing using double discrete wavelet transform. computer vision and pattern recognition pp. 1091–1098 (2013)
27. Zhao, J., Feng, H., Xu, Z., Li, Q., Tao, X.: Automatic blur region segmentation approach using image matting. Signal, Image and Video Processing **7**(6), 1173–1181 (2013)
28. Zhao, W., Zhao, F., Wang, D., Lu, H.: Defocus blur detection via multi-stream bottom-top-bottom network. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
29. Zhao, W., Zheng, B., Lin, Q., Lu, H.: Enhancing diversity of defocus blur detectors via cross-ensemble network. Computer Vision and Pattern Recognition pp. 8905–8913 (2019)
30. Zhu, X., Cohen, S., Schiller, S.N., Milanfar, P.: Estimating spatially varying defocus blur from a single image. IEEE Transactions on Image Processing **22**(12), 4879–4891 (2013)
31. Zhuo, S., Sim, T.: Defocus map estimation from a single image. Pattern Recognition **44**(9), 1852–1858 (2011)