IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 21, 2020, accepted February 18, 2020, date of publication March 3, 2020, date of current version March 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978084

Automatic Extraction of Blur Regions on a Single Image Based on Semantic Segmentation

AODONG SHEN[1,2], HAN DONG[1], KUN WANG[1], YOUYONG KONG[1,2], JIASONG WU[1,2], (Member, IEEE), AND HUAZHONG SHU[1,2,3], (Senior Member, IEEE)

[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing 210096, China
[3]International Joint Laboratory of Information Display and Visualization, Nanjing 210096, China

Corresponding author: Youyong Kong (kongyouyong@seu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876037, in part by the State's Key Project of Research, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT Blur region detection from a single image with spatially-varying blur is a challenging task. Although many methods are proposed in the past decades, most of them are based on hand-crafted features. These features are not robust to image context, image size, blur type and other factors, which cannot obtain sound performance. In addition, the craft of these features requires a lot of domain knowledge. To address these problems, in this paper, a blur region detection method based on semantic segmentation is proposed to extract blur regions, which well integrates global image-level context and cross-layer context information making the auto-learned features more robust. Specifically, we design a blur detection net (BDNet) for blur detection by combining ResNets and FCNs. A binary mask can be produced in an end-to-end way. By our method, the mean region intersection over union (Mean IoU) increased by nearly 20% compared with most other blur detection methods. We make the code publicly available at *https://github.com/SEU-DongHan/BDNet.*

INDEX TERMS Deep learning, convolutional network, blur detection, semantic segmentation.

I. INTRODUCTION

Blur can be regarded as one type of photo degradation caused by many factors in the process of photo acquisition or post-processing. Blur leads to the loss of details and prevents people from obtaining the information of the real scene captured by the image quickly and accurately. There are two common types of blur phenomenon, named motion blur and defocus (out of focus). The former is mainly caused by camera relative motion during exposure and the latter is caused by lens aberrations [1]. With the rapid development of computer vision techniques, image content understanding has become a hot research field, and thus, it becomes essential to uncover the information immersed in the blurred image. Actually, efficient and effective extraction of blur regions can naturally benefit many applications, including scene classification, object detection, image quality assessment, image restoration [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

In this paper, we will mainly focus on the methods to solve the spatial-varying blur detection problem. A number of algorithms have been proposed over the past decades to deal with this problem. Most of the existing methods are based on hand-crafted features [2]–[4]. Inevitably, the performance of these methods mainly depend on the discriminative ability of these features. However, the craft of robust features is not an easy task, because the image context, image size, and other factors may have a large impact on them. Moreover, most of the previous methods only utilize the local information from the image patch [4], [5]. Therefore, to get dense blur map, the size and stride of the sliding window should be well trade-off. Using a local image patch cannot capture the large context information which will be helpful for the classification of texture flatten areas in blur detection problems. Last but not least, some methods are proposed to address the blur detection problem of a specified blur type [6]–[9]. In this case, the blur type is seen as a kind of prior knowledge, which is not always available. In addition, a blurred area of a spatial-varying blur image may be blurred by mixed of blur

kernels. In such cases, this kind of methods may not work well.

In recent years, deep learning has become a promising framework, which has been widely applied in the fields of computer vision, pattern recognition and signal processing [10]–[12]. A variety of effective networks have been proposed for different applications. VGG [13], Google-Net [14], ResNet [15] and etc. are put forward for image classification. FCN [16], SegNet [17], UNet [18] and etc. are proposed for semantic segmentation. The problem to be solved in this paper is to distinguish blur regions that can be detected by human eyes. Given an image, a binary mask should be generated to show each pixel is blur or not, which can be regarded as a classification problem in pixel level. From this point of view, there is much similarity between blur detection and semantic segmentation. Therefore, the approaches for semantic segmentation can be adopted to process blur detection problem and this is the key insight of our approach. However, the blur detection is different from the common object semantic segmentation task. Firstly, due to the existing of the gradual blur, the boundary between blur and clear regions may be not sharp and is hard to determine. Secondly, there is no massive and public dataset available.

To overcome the aforementioned two issues, we propose a novel deep neural network called BDNet by fusing the ResNets and FCNs. Following the common network structure for semantic segmentation, we use ResNets as our encoder part and FCNs as decoder part. We further extend FCNs to FCN2s. Moreover, instead of training the model stage by stage, we train the model all at once. To make the progress more stable and efficient, the batch normalization layer is added before the merge of features from different layers. We also make some other improvements to the structure of the combined network. With limited data, the technique of transfer learning is employed to get better results and save the time of training. Finally, by the proposed method, the blur regions can be detected successfully and a competitive result is reached.

The proposed method has several advantages over other existing spatial-varying blur detection methods on a single image. First of all, with the powerful auto feature extraction ability of deep learning framework, it is unnecessary to craft features by domain knowledge. It is shown that these auto-extracted features are more robust to image size, the image context and etc. Secondly, differing from image patch based approaches, with relatively large reception field of view, our proposed framework can capture the context information efficiently. A better segmentation result can be achieved with a small noise. Thirdly, the fully convolution network structure can process different sizes of images. Fourthly, just a single image is used to get segmentation results, which makes our method more practical. Compared with other classical methods, the mean region intersection over union (Mean IoU) [16] of our method increases by about 20%.

The organization of the result of the paper is as follows. In the next section, we review related work on blur detection, deep neural networks and semantic segmentation based on convolutional neural network. In Section 3, we firstly introduce the datasets and then present the proposed approach and implementation details. Section 4 gives the experimental results, where we compare the proposed results with some state-of-art methods. Finally, we conclude the paper in Section 5.

II. RELATED WORK

A. BLUR DETECTION

Many blur detection methods are proposed in the past few decades. According to the input, these blur detection methods can be categorized into sequence-image based and single-image based ones. Sequence-image based algorithms can be classified into three types, frame difference based, background modeling based and decomposition based ones [3]. Single image blur detection algorithm can be further classified into spatial-uniform and spatial-varying ones. As the name suggests, for spatial-uniform blur detection problem, there is only one type of blur and the blur strength does not vary in spatial, while spatial-varying is opposite. However, spatial-varying images are more common, most of the recent research is conducted on this kind of images. So, we will mainly review related works on this problem.

Spatial-varying blur detection is a much more challenging task compared with spatial-uniform blur, because blur varies in different blur regions and different blur types present. Actually, blur detection can be accomplished by blur-nonblur classification or by blur estimation. The former attempts to determine whether or not a region is blur. The later can tell the blur kernel straightly or just estimate blur strength. Among the later, there is one kind of estimation algorithms concentrating on constructing discriminative blur features directly for input images and make them potent enough to differentiate between different strength of blur. By mining blur cues of the input image, many features have been proposed especially for blur detection task, most of which are handcrafted and derived from spatial domain, frequency domain, color space and etc. Su *et al.* [19] detect blurred image regions by examining singular value information for each image pixels. Shi *et al.* [20] propose a simple yet effective blur feature to detect and estimate just noticeable blur (JNB) via sparse representative and image decomposition. Rather than using a single feature, some learning-based methods try to combine several features together with classifiers. Liu *et al.* [4] develop several features modeled by image color, gradient and spectrum information, and then a Bayesian classifier is trained to discriminate whether or not an image patch is blur. Shi *et al.* [2] study a few blur feature representations in image gradient, Fourier domain, and data-driven local filters to differentiate between blurred and clear image regions. Yi *et al* [53] propose a sharpness metric based on Local Binary Patterns (LBP) to separate blur and non-blur regions.

Instead of just mining discriminative features, the other kind of algorithms focus on further fitting the relationship between blur kernels and blur features or estimating the blur kernels straightly. However, different blurs are modeled differently, most of this kind of algorithms are designed for specified blur type. Actually, blur kernel estimation is the first step of most of deblur algorithms [21]–[23], these methods can be easily adapted to blur detection. Many task-specific methods are proposed to estimate the blur kernel for blur detection problem. Chakrabarti *et al.* [5] apply shot-time Fourier transform on directional blur image patches and find a local blur cue that measures the likelihood of a small neighborhood being blurred by a candidate blur kernel. Pang *et al.* [3] propose a new kernel-specific vector with the information of a blur kernel and an image patch and classifiers are used to estimate the blur kernels. To deal with defocus blur images, Zhuo *et al.* [6] re-blur the input image and then blur kernel is estimated at edge locations, full blur map is achieved by propagation techniques. Tang *et al.* [9] establish the relationship between the amount of defocus blur and spectrum contrast at edge locations, similar propagation operation is used to obtain full blur map. Zhang *et al.* [7] model the blur kernel at edge location as Gaussian function, which is estimated by combing the gradient information of input image and the re-blurred ones, and then K nearest neighbors (KNN) [24] matting interpolation is used to get full blur map after the sparse blur map is refined by guided filter [25]. Golestaneh and Karam [48] propose a method to compute blur detection maps based on High-frequency multiscale Fusion and Sort Transform (HiFST) of gradient magnitudes to determine the level of blur at each location in an image.

However, the aforementioned methods have some disadvantages. Firstly, most of the previous works are based on observation or handcrafted features. The craft of a feature with high discriminative ability and generality is still not an easy work, which may require much domain knowledge. It is most possible that these methods are not robust enough, easily affected by the context of image, image size and other factors. Secondly, most of these features are crafted only using the local information of an image patch, image global context information could not be captured effectively, and therefore the texture flatten region is not handled appropriately. Thirdly, some algorithms are designed for some specific blur type which is a kind of prior information not always available.

There are also some methods try to analyze blur images by the deep neural network. Yan and Shao [26] apply pre-trained deep neural network (DNN) and general regression neural network (GRNN) to the problem of blur analysis for the first time. Sun *et al.* [27] propose a deep learning approach to address the problem of estimating and removing non-uniform motion blur from a single image. They first predict the probabilistic distribution of motion blur at the patch level. Then a Markov random field model is used to infer a dense non-uniform motion blur field. Finally, motion blur is removed by a non-uniform deblurring model using patch-level image prior. To address the problem of segmenting dynamic objects, Punnappurath *et al.* [28] develop a CNN to predict the probabilistic distribution of the composite kernel, and then automatically segmentation is performed at each depth layer. Kim *et al.* [52] propose a network with long residual skip-connections and multi-scale reconstruction loss functions to detect blur regions. Basically, these deep learning models are trained to learn blur kernels. But unlike deep learning based methods above, we tackle blur detection problem by telling a pixel is whether blur or not, or estimating the blur strength.

B. SEMANTIC SEGMENTATION

Li *et al.* [49] propose a method called vector field convolution (VFC) to segment images. The VFC field can be calculated by convolving a vector field kernel with the edge map generated from the image. This method is less computationally expensive and robust to noise, but the weak edges in images might be overwhelmed by the strong edges along with the noise. Maghsoudi *et al.* [50] use superpixels to segment images for animal morphology research. The proposed method uses SLIC to remark and segment images, but the speed and accuracy still need to be improved. At present, deep learning is mostly used for image semantic segmentation compared to traditional methods.

The rapid development of deep convolutional network is inspired by AlexNet [29] as a winner of the ILSVRC 2012 competition. After that, growing numbers of participators try to develop novel and effective networks. VGG-nets [13], GoogleNets [14] win the championship of ILSVRC successively. Recently, the winner of ILSVRC 2015, He *et al.* [15] propose a much deeper convolutional network called ResNet presenting a good solution to the training problem of deep convolutional networks and showing the better auto-feature learning ability of deeper networks. Later, Zagoruyko and Komodakis [30] propose a novel architecture called wide residual network by decreasing depth and increasing width of residual networks. Xie *et al.* [31] present ResNext exposing a new dimension of convolutional neural network.

Driven by deep learning, significant progress has been made in semantic segmentation. A number of frameworks are proposed based on deep convolutional neural network. Among them, FCN is the most classical and successful one. The key insight of FCN is to build ''fully convolutional'' networks that produce output with the same size as input. It can be trained end-to-end. Skip architecture is employed to combine semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentation [16]. Later, some variation networks are developed. DeepLab models [32] enlarge the reception field of view by dilated convolution and use conditional random field(CRF) model as a post-processing tool to refine segmentation result. Deconvolution networks [33] use a stack of learnable deconvolution and upsampled layers to restore segmentation resolution. UNet [34] combines skip

FIGURE 1. The first row is the original images in the dataset, and the second row is the corresponding ground truth images. The ground truth marks the blur area as white and non-blur area as black.

layers and learnable deconvolution layers for pixel labeling. PaseNet [35] employs average pooling layer to capture global image context information and normalize feature maps before fusion. Segnet [36] uses the pooling indices computed in the max-pooling step to perform non-linear upsampling.

Blur detection can be accomplished by blur estimation or by blur-nonblur classification. The former needs to mine blur features, most of which are handcrafted and derived from spatial domain, frequency domain and etc. However, deep learning methods can omit the processes of extracting features manually. Deep learning methods can transform the blur-nonblur classification into semantic segmentation, and obtain more abstract features by training networks. Extracting blur regions by semantic segmentation can be more accurate and robust.

III. DATA AND METHOD

A. DATASETS

There are three datasets used in our experiments called BlurRaw, BlurDB1 and BlurDB2. The BlurRaw dataset is contributed by Shi *et al.* [2] containing 1000 annotated images with 296 motion blur and 704 defocus. Some sample images are shown in Fig. 1. Here, we find some annotation errors (out_of_focus0508, etc.) and have fixed them. These images vary in size. We find the mean height of these images is 467 and the mean width is 617. But the maximum width reaches 1024.

Based on BlurRaw, we generate two datasets called BlurDB1 and BlurDB2. To generate BlurDB1, we randomly crop image patches with a size of 256×256 from raw images. To get the same number of images for different blur types, we crop 12 image patches from each motion blur image and 5 image patches from out of focus image specifically. Finally, we obtain 7072 images in total, with 6727 training images and 345 testing images. The details are shown in Table 1. BlurDB2 is generated to make the best of limited data. We crop image patches from top left to bottom right with 1/2 area overlap for every two adjacent patches. Finally, we get 9134 images in total. Same as the proportion of training set and test set used in current research [51], 10% of the dataset are used for testing and 90% for training. More details are shown in Table 2.

TABLE 1. Details about BlurDB1. There are 7072 images in BLURDB1, of which 6727 are used as training images and 345 are used as testing images.

Data	Motion/Raw	Defocus/Raw	Total
Train	3372 / 281	3355 / 671	6727 / 952
Test	180 / 15	165 / 33	345 / 48
Total	3552 / 296	3520 / 734	7072 / 1000

TABLE 2. Details about BlurDB2. There are 9134 images in BLURDB2, of which 8215 are used as training images and 919 are used as testing images.

Data	Motion/Raw	Defocus/Raw	Total
Train	2382 / 265	5833 / 635	8215 / 900
Test	292 / 31	627 / 69	919 / 100
Total	2674 / 296	6460 / 734	9134 / 1000

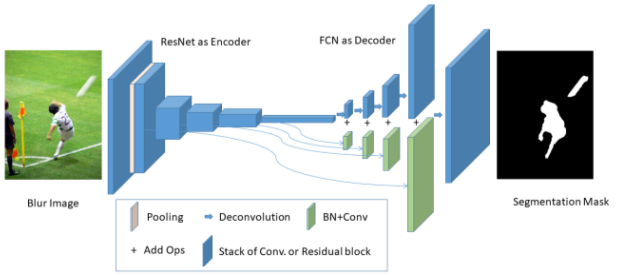

FIGURE 2. BDNet network structure. The ResNet used as Encoder to extract features and the FCN used as Decoder to get results.

B. BDNet-NETWORK STRUCTURE

Our general network called BDNet is shown in Fig. 2. Our network fuses ResNets and FCNs for transfer learning. FCNs is an effective general framework for semantic segmentation. And ResNets have outperformed other classification networks in many tasks. In our proposed network structure, ResNets are used as encoder to extract features and FCNs are used as decoder to get dense segmentation results. With transfer learning, we train the network to detect blur regions successfully.

1) DEEP ResNet

Deep neural networks have shown impressive performance in many tasks. The key point is to design a novel network structure. It is evidenced that the network depth is of crucial importance on the feature representations [14]. After VGGs and GoogleNets, a growing number of people focused on the design and training of deep neural networks. However, simply increasing the depth of neural networks does not improve anymore, which is known as degradation problem. He *et al.* [15], [37] propose a deep residual learning framework to handle this problem and it shows compelling results and nice convergence behaviors on ImageNet and MS COCO competitions. It is revealed that by using identity mappings as the skip connections and after-addition activation, the forward and backward signals can be directly propagated from one

block to any other block without adding extra parameters or computation complexity.

Generally, the residual unit with identity mapping performs the following computations:

$$x_{l+1} = h(x_l) + F(x_l, W_l) \tag{1}$$

where x_l is the input feature to the l-th residual unit and W_l is a set of weights associated with the l-th residual unit. $h(x_l)$ is an identity mapping. F denotes the residual functions to be learned, e.g. a stack of convolution layers. After unfold the formula recursively, we will have:

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i, W_i) \tag{2}$$

The feature x_L of any deeper unit L can be represented as the feature x_l of any shallower unit l plus a residual function with a form of $\sum_{i=1}^{L1} F(x_i, W_i)$. This leads to nice backward propagation properties. From the chain rule of back propagation, we have:

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, W_i)) \tag{3}$$

where ε denotes the loss function. The formula indicates that the additive term of $\partial\varepsilon/\partial x_L$ can be propagated directly back to any shallower unit l.

2) FULLY CONVOLUTIONAL NETWORK (FCN)

A standard deep CNN architecture for whole-image classification typically consists of a stack of convolution layer, nonlinear activation function, pooling layer, and fully connected layer. If no adaptive pooling operations are used, the input of fully connected layer must have fixed dimensions. Long *et al.* [16] cast the classifier networks to full convolutional networks that can take an input of any size and output classification maps by replacing full-connected layers with convolutional ones. However, the repeated combination of max-pooling and downsampling layers results in feature maps with a significantly reduced spatial resolution [32]. To overcome this problem, FCNs use in-network upsampling layers (deconvolutional layers initialized by bilinear upsampling) to produce a dense prediction. Dense prediction can also be yielded by the trick of input shifting and output interlacing, but Long *et al.* find that learning through upsampling is more effective and efficient. To obtain accurate and detailed segmentation, a skip architecture is introduced combining semantic information from deep layer to shallow layer, which improves the performance of FCNs dramatically.

Most of the previous blur detection methods use image patches. However, the patch-based processing flow is wasteful and time-consuming [38]. The estimated blur strength, kernels or the binary labels are corresponding to the center pixel of a cropped image patch. To obtain full blur map, the stride of the sliding window should be small enough. Unavoidably, this will result in some redundant computing operations between neighboring patches. Moreover, the size of image context is determined by image patch, which should be bigger enough to make more information available for better estimation or classification result. But bigger patch size will enlarge the unhandled area at the borders of images. And patch size also has some influence on the performance of some features. So, the image size should be traded off critically for these methods.

On the contrary, our framework can produce dense prediction with a large reception of field of view without redundant operations, which is very helpful to handle texture flatten regions.

3) NETWORK FUSION AND TRANSFER LEARNING

Here, we extend FCNs to FCN2s to make the best of the skip connection structures. More low-level features can be fused to final results. We think it is more important for semantic segmentation tasks without sharp segmentation boundaries like blur detection. In order to train the network all-at-once without divergence, scale layers proposed by [16] with fixed constants picked artificially are used before feature map fusion from different layers. However, the scale of features from different layers may be quite different, which makes it difficult to combine directly the features for prediction [35]. Therefore, the scale layer is replaced with batch normalization layer in BDNet, which makes it easier to train and avoid picking constants. In VGGs, the downsampling operations are implemented by max pooling layers. But, except max pooling layer 1, the downsampling operations in ResNet are performed by convolution layer with stride 2. Thus, in our experiments, skip connection is added after each block with downsampling operation or concatenation layers after downsampling block. Moreover, deconvolution layer is utilized to upsample, and dropout layer is employed at the top of ResNet to avoid over-fitting and this leads to a better result.

Transfer learning can apply knowledge previously learned in one domain to another. Generally, transfer learning can use previously trained model for a new task, and the model usually is trained on large datasets. Transfer learning can save a significant amount of training effort [39]. With limit data, this technique provides a significant head start toward learning appropriate visual features. Specifically, we use the model pre-trained on ImageNet [40], [41] data to initial encoder part (ResNet) of our network. We also try two transfer learning methods in our experiments. First, the parameters of ResNet part is fixed without updating in the phase of training. Second, the entire network is trained jointly with low learning rate in encoder part, and achieves better results. With this strategy, our network spends less time to train and gets better results with limited annotated data.

C. IMPLEMENTATION DETAILS

Most of the models in this paper are trained and tested with Caffe [42] on a single NVIDIA GTX 1070. The parameters mainly refer to [16], which are general parameters that used in the relevant research. We use SGD as optimizer with the mini-batch size of 5. The weight decay is 0.0005 and

the momentum is 0.9. we fine-tune our network using pre-trained weights, start from learning rate at 1e-3 and divide it by 10 for three times at step 21000, 42000, 56000 with the max iteration 70000. All deconvolution layer in FCNs initialized with bilinear kernels and other layers with Xavier [43] random variables. The weight used in VGG16 series experiments is provided by [16](FCN-32s) fine-tuned from the ILSVRC-trained VGG16 Model and the VGG16 networks also follow the same structure with [16]. The ResNets network structure is implemented following [37], and the code is publicly available with pre-trained weights. The cross-entropy loss is utilized as the objective function.

The input of the network is a 224 × 224 color image with RGB three channels. The input images are preprocessed by subtracting the mean value in each channel. The training data are augmented by randomly cropping, flipping horizontally or vertically. Finally, we trained the model 100 epochs for 7 hours and 55 minutes in our experimental environment.

IV. EXPERIMENTS

A. EVALUATION METRICS

Blur detection is often evaluated by visual inspection and quantitative assessments with Precession-Recall (PR) Curve and Receiver Operating Characteristic (ROC) Curve. In addition, four novel metrics are introduced from common semantic segmentation and scene parsing evaluations to compare methods fairly, which are the variations on pixel accuracy and region intersection over union, namely pixel accuracy, mean accuracy, mean IoU, frequency weighted IoU.

Pixel accuracy is a commonly utilized measurement which represents the percentage of total pixels that have been classified correctly. Mean accuracy calculates the proportion of pixels correctly classified in each class and then averages the results of all classes. Mean IoU stands for mean intersection over union and is a standard measure of semantic segmentation. It can calculate the ratio of the intersection and union of two sets, which in the case of semantic segmentation are ground truth and predicted segmentation. Frequency weighted IoU sets the weight for each class according to the frequency of occurrence.

The metrics above can be computed by flowing formulations:

Pixel Accuracy:

$$(\sum_i a_{ii})/(\sum_i t_{ii}) \tag{4}$$

Mean Accuracy:

$$(1/a_c)\sum_i (a_{ii}/t_i) \tag{5}$$

Mean IoU:

$$(1/a_c)\sum_i (a_{ii}/(t_i + k_i - a_{ii})) \tag{6}$$

Frequency Weighted IoU:

$$(\sum_k t_k)^{-1}\sum_i (_i n_{ii}/(t_i + k_i - a_{ii})) \tag{7}$$

where, $\boldsymbol{\alpha_{ii}}$ denotes the number of pixels labeled by i and predicted to i. The number of classes $\boldsymbol{\alpha_c}$. $\boldsymbol{t_i} = \sum_j \boldsymbol{\alpha_{ij}}$ denotes the total number of pixels of i. $\boldsymbol{k_i} = \sum_j \boldsymbol{\alpha_{ji}}$ denotes the total number of pixels predicted to i.

B. COMPARISON WITH OTHER METHODS

The proposed method is compared with other existing blur detection methods of Shi *et al.* [2], Liu *et al.* [4], Chakrabarti *et al.* [5], Zhang *et al.* [7], Su *et al.* [19], Golestaneh and Karam [48], Kim *et al.* [52] and Yi *et al.* [53]. The methods of Shi *et al.* [2], Liu *et al.* [4], Su *et al.* [19], and Yi *et al.* [53] can mine discriminative features of images. The methods of Chakrabarti *et al.* [5], Zhang *et al.* [7] and Golestaneh and Karam [48] focus on further fitting the relationship between blur kernels and blur features or estimating the blur kernels straightly. The method of Kim [52] uses deep neural network to extract blur regions. Here we use their code publicly available to produce the results. The results of all these methods are in the form of blur map. For a fair comparison, we modify the output layer of the model and use sigmoid function to get blur map instead of binary results. The data for evaluation comes from database BlurRaw, we use 10% raw images left for test. Our model used here is BDNet (with BN, Dropout) trained on BlurDB2.

In the experiments, we used 10% of training set for validation and trained the model with 100 epochs. The loss curves of train and validation are shown in Fig. 4. The trends of two curves are consistent, which can demonstrate that the model is not overfitting. In addition, we also performed k-fold cross validation and trained model with 100 epochs, where k is set to 3. The experiment results are shown in Fig. 5. The results show that the performance between different folds is quite similar, especially after about 60 epochs. The experiments further demonstrate the robustness and generalization ability of the proposed method.

The visual comparison is shown in Fig. 3. We see that our method can obtain competitive results. The overall relatively poor performance of other methods accounts for many reasons. Firstly, noise is not well controlled. Secondly, some methods can only capture edge information which results in many holes leading to more misclassification. Thirdly, texture flatten areas are not handled appropriately. In addition, we can find through the PR Curve and ROC Curve in Fig. 6 and Fig. 7 that our method performs much better than other methods.

The quantitative comparison is shown in Table 3. As we can see from the table, our method outperforms all other methods in terms of all metrics. Our method achieves a mean IoU of 0.800, pixel accuracy and mean accuracy over 0.9. And moreover, the mean inference time of our method is less than 10 seconds with our computer, which does not take more time than Zhang [7]. And what is more important is that no post-processing methods are applied in our method.

We also find some pictures do not achieve expected results using our method such as images 7-10 shown in Fig. 3. A large number of pixels are misclassified when a large

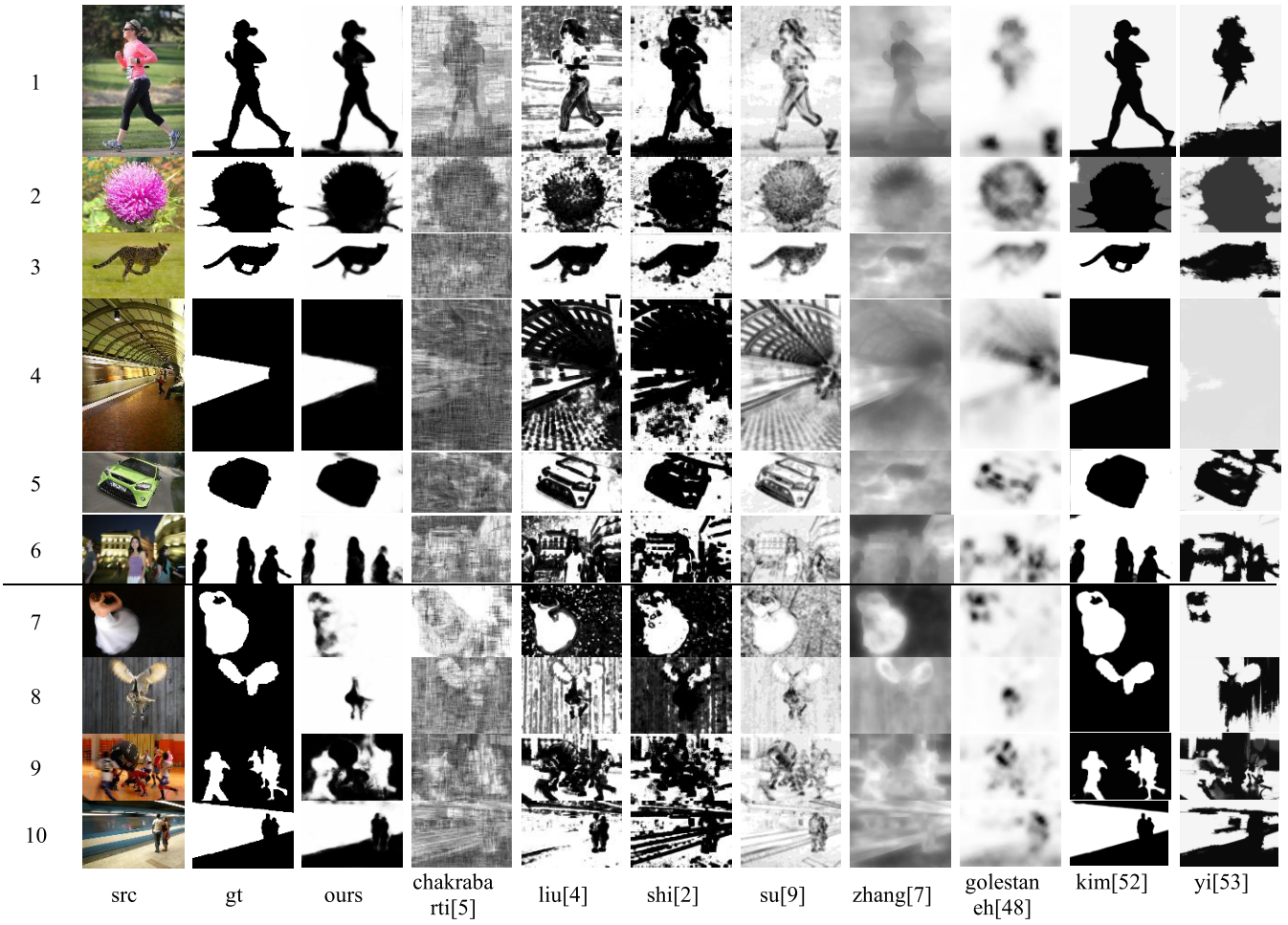

FIGURE 3. Detection results of different methods, "src" is the blur image, "gt" is ground truth. The figure shows intuitively that our method obtains competitive results.

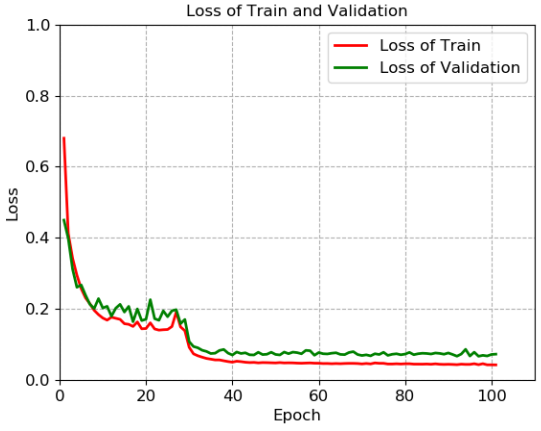

FIGURE 4. The trends of training loss curve and validation loss curve are consistent, which can prove our model is not overfitting.

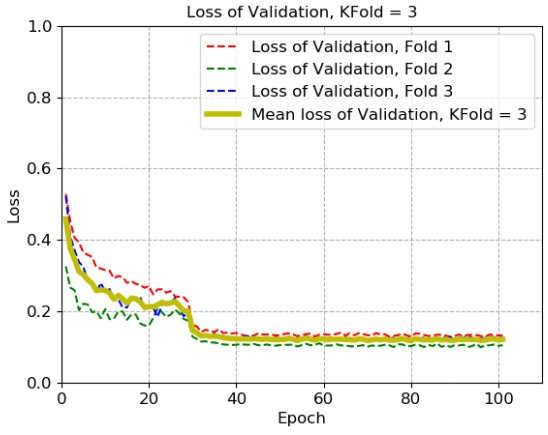

FIGURE 5. The performance between different folds is quite similar. It further demonstrates the robustness and generalization ability of the proposed method.

texture flatten area presents. Our model is trained on small sized images, which has some impact on our results. We think this problem can be solved by fusing more global context information or training on datasets with larger sized images. Another problem is that the boundary is not determined accurately if the blur boundary is not sharp enough, especially for motion blur images. And the fact that the number of annotated motion blur images is smaller than out of focus images makes it worse. In fact, considering the imaging mechanism, the scenes captured by defocus images are static. Although the blur amount may change continuously in some defocus images, the blur amount of most of this kind of

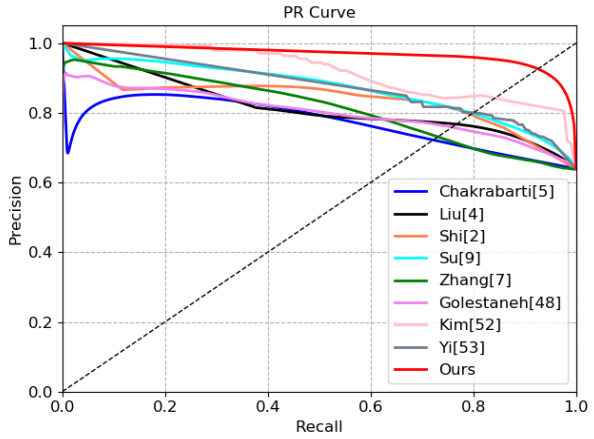

FIGURE 6. Precession-Recall (PR) Curve of different methods. The red curve on the top represents our method and the results are better than the other methods.

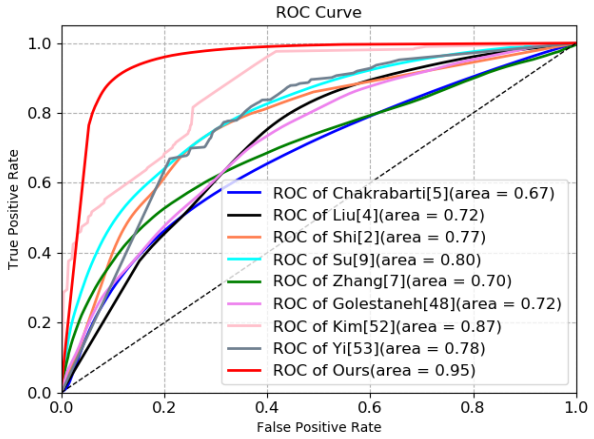

FIGURE 7. Receiver Operating Characteristic (ROC) Curve of different methods. The red curve on the top represents our method, with the Area Under The Curve (AUC) of 0.95, which is higher than the other methods.

TABLE 3. Results of other methods and our method. The results of our methods in four metrics are better than others.

Method	Pixel Accuracy	Mean Accuracy	Mean IoU	Frequency Weighted IoU
Ours	**0.904**	**0.901**	**0.800**	**0.849**
Chakrabarti[5]	0.421	0.445	0.329	0.362
Liu[4]	0.590	0.600	0.440	0.498
Shi[2]	0.625	0.691	0.487	0.530
Su[9]	0.701	0.674	0.541	0.611
Zhang[7]	0.647	0.691	0.477	0.538
Golestaneh[48]	0.764	0.738	0.583	0.653
Kim[52]	0.858	0.873	0.784	0.808
Yi[53]	0.750	0.765	0.584	0.648

images changes abruptly. So, the blur boundaries of defocus images are sharper than motion blur images. That is very helpful for blur detection. Relatively, it is more difficult to solve the problem of motion blur detection. Due to the relative motion between camera and objects, it is hard to determine where the motion area starts and ends. The available

TABLE 4. Results of VGG16-FCN32s-OL and VGG16-CN32s. The results show that online learning yields better model.

Model	Pixel Accuracy	Mean Accuracy	Mean IoU	Frequency Weighted IoU
VGG16+FCN32s+OL	**0.893**	**0.858**	**0.754**	**0.836**
VGG16+FCN32s	0.885	0.831	0.730	0.821

boundary information is very limited which may confuse our model and leads to bad performance, even we have fused lower level features by extending FCN to FCN2s. Moreover, images with RGB channels may be helpful for segmentation tasks with sharp edges such as object segmentation, but not good at blur detection. So this problem may be improved by some sample instance type aware learning techniques or some post-processing methods such as CRF(Conditional Random Field) and cluster [45]. Last but not least, massive data are always the base stone of deep learning, and we argue that more training data will definitely be helpful for getting better results.

C. EXPERIMENTS WITH FCNs AND ResNets

In this section, we first investigate the capacity of FCNs based on VGG16, and then conduct comprehensive experiments on BDNet. The models in a), b) and c) are trained on BlurDB1 and d) is on BlurDB2. In all of the experiments of a), b) and c), VGG16 models follow the same structure with paper [16]. The differences of comparison results in each experiment may be small because of the similar overall structure of the network.

1) ONLINE LEARNING STRATEGY

The paper [16] empirically shows that online learning (high momentum, one example for each step) works well and yields better models. To verify that, we train two models VGG16-FCN32s-OL and VGG16-CN32s, the former is trained by the strategy employed by the paper and the latter is trained by the way aforementioned in section 3.3. The results are shown in Table 4. It shows that the empirical strategy does work well, the former model outperforms the latter by 2.4% in term of mean IoU. But we think that this strategy may be suitable for traditional networks without batch normalization. For networks where batch normalization layer is widely used, it may become unsuitable. And this train strategy cannot make full use of hardware computing power and takes more time. So, the training strategy employed hereafter by following experiments is the same as what is described in section 3.3.

2) FROM FCN32s to FCN2s

As stated in the paper [16], the success of FCNs is mainly contributed by skip structures. We train a series model from FCN32s to FCN8s and further extend to FCN2s. And here, we train these models stage by stage which means that FCN16s is fine-tuned from FCN32s, and then FCN8s is fine-tuned from FCN16s. The results are shown in Table 5 and we can see that the results are getting better with more

TABLE 5. Results of fcnxs trained stage-by-stage. The table shows that results are getting better with more skip layers added.

Model	Pixel Accuracy	Mean Accuracy	Mean IoU	Frequency Weighted IoU
VGG16+FCN32s	0.885	0.831	0.730	0.821
VGG16+FCN16s	0.891	0.856	0.754	0.834
VGG16+FCN8s	0.890	0.858	0.755	0.833
VGG16+FCN4s	0.891	0.858	0.756	0.834
VGG16+FCN2s	**0.894**	**0.859**	**0.762**	**0.838**

TABLE 6. Results of models trained by all-at-once strategy. scale layer replaced by batch normalization layer achieves better results.

Model	Pixel Accuracy	Mean Accuracy	Mean IoU	Frequency Weighted IoU
VGG16-FCN8s-Once	0.902	0.865	0.771	0.849
VGG16-FCN8s-Once-BN	**0.905**	**0.868**	**0.775**	**0.852**

skip layer is added. Model VGG16-FCN2s achieves the best results in terms of all metrics, specifically which outperforms VGG16-FCN32s by 3.2% on mean IoU. It suggests that FCN2s can still improve the results for blur detection problem especially in the case where the segmentation boundaries are not sharp enough.

3) ALL-AT-ONCE TRAINING STRATEGY

The paper [16] also shows another all-at-once training strategy, which is faster and less tedious and implemented by scale layers in each stream by a constant picked to approximately equalize average feature norms across streams. Instead of using a layer with fixed constant, we propose to replace it with batch normalization layer to avoid the setting of the constants. We get a minor improvement as shown in Table 6, which verifies the feasibility of our proposal.

4) RESULTS WITH OUR IMPROVEMENTS

We also try to do some improvements to ResNet to achieve better performance as mentioned in the previous section and different methods of transfer learning. Model BDNet(w/o BN) is fine-tuned without batch normalization layer in skip connections with fixed parameters of ResNet, and then batch normalization layer is added to BDNet(with BN). At the same time, we train the both two parts of ResNets and FCNs jointly. Finally, dropout layer is added to BDNet (with BN, Dropout). The Table 7 shows that batch normalization layer, joint training and dropout layer are helpful. BDNet (with BN, Dropout) increases mean IoU by 1.4% on BlurDB2 and 1.8% on BlurRaw compared with BDNet(w/o BN).

D. INFLUENCE OF PARAMETERS

1) DO THE DIFFERENT TYPES OF BLUR IMAGES INFERENCE WITH EACH OTHER

In order to explore the influence, we train two different models respectively on the two kinds of degraded image sets in BlurDB2. The models in this section are trained by Pytorch [46] with similar settings. We find that BDNet

TABLE 7. Results of BDNet(w/o BN), BDNet(with BN), BDnet(with BN, Dropout). The table shows that batch normalization layer, joint training and dropout layer are helpful to improve the results.

Data	Model	Pixel Accuracy	Mean Accuracy	Mean IoU	Frequency Weighted IoU
BlurRaw	BDNet(w/o BN)	0.892	0.886	0.782	0.831
	BDNet(with BN)	0.898	0.896	0.792	0.840
	BDNet(with BN, Dropout)	**0.904**	**0.901**	**0.800**	**0.849**
BlurDB2	BDNet(w/o BN)	0.894	0.873	0.780	0.840
	BDNet(with BN)	0.900	0.877	0.786	0.849
	BDNet(with BN, Dropout)	**0.901**	**0.882**	**0.794**	**0.851**

TABLE 8. Results of BDNet-motion, BDNet-Defocus, BDNet-ALL. The table shows that training bdnet with mixed degraded image sets will not interfere results too much.

Model	Pixel Accuracy	Mean Accuracy	Mean IoU	Frequency Weighted IoU
BDNet-Motion	0.864	0.856	0.745	0.813
BDNet-Defocus	0.930	0.910	0.838	0.886
BDNet-ALL	0.913	0.895	0.812	0.867
Weighted Average	0.909	0.893	0.809	0.863

achieves better performance, which is because of the bigger batch size and more flexible training strategies. In details, we train these models using SGD optimizer with learning rate 0.001, momentum 0.9 and batch size 16 for 100 epochs on BlurDB2. We drop the learning rate by 0.1 in every 30 epochs. The decoder parts of these models are initialized by the weights pre-trained on ImageNet and the other parameters are initialized by Xavier. We firstly fix the pre-trained weights and then train the whole network jointly.

The results are shown in Table 8. The mean IoU of the model (BDNet-Motion) trained with only motion blur images is up to 0.745, which is lower than the model (BDNet-ALL) trained on the whole dataset (0.812). The mean IoU of the model (BDNet-Defocus) trained with only defocus images reaches 0.838. As shown in Table 2, the motion blur images take about 1/3 part of the whole test set of BlurDB2. If we apply a proportionally weighted sum to the mean IoU of the two models, we will find it roughly equals to the result of the model trained on the whole dataset. So, we can make a conclusion that it will not interfere with each other too much if we train BDNet with the mixed degraded image sets. Actually, there are some defocus areas in an image with motion blur, it does make sense to train the model with the mixed data.

2) COMPARISON WITH OTHER DEEP NEURAL NETWORKS

Here we try to compare with some state-of-art methods. We notice that the saliency detection problem is much similar to blur detection, and they have the same kind of inputs and outputs. Here we compare our method with DSS [47] proposed for saliency detection, SegNet and UNet presented for semantic segmentation. DSS is trained with Caffe as described in the paper [47] using the code publicly available.

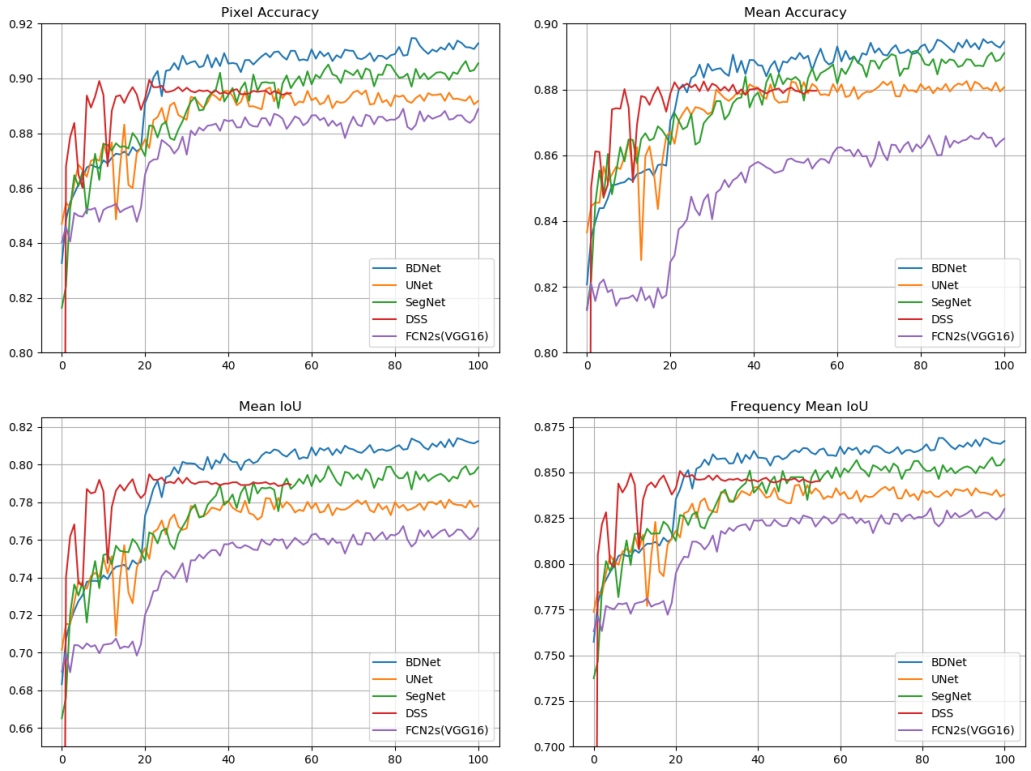

FIGURE 8. The results of BDNet(ours), UNet, SegNet, DSS, FCN2s(VGG16). The horizontal axis in each figure represents the 'Iteration', and the vertical axis represents the results of each metric.

The other models in this section are all trained by Pytorch [46] with similar settings like the last section.

The results are shown in Fig. 8. DSS adds more short connections and supervised losses. We can see that they are very helpful, the metrics of segmentation results increase quickly at the beginning. But DSS does not get a good result finally. We also find it is very time-consuming to train DSS and here we only train 56 epochs. UNet's operation brings more information to the decoder part but it seems that add operation used by FCN is more straightforward and give a more powerful guidance to the decoder part. SegNet performs better by a kind of special upsample method, but modern convolutional neural networks downsample the size of feature map not only by pooling layer but also by convolutional layer and e.t. It is difficult for SegNet to extend these neural networks. In general, our method gets a good result in a simple yet effective way.

V. CONCLUSION

We present a novel method to solve the blur detection problem. Compared with other methods, the main advantages of our method are as follows: 1. Our method is robust and has good segmentation results with a little noise. 2. Our method can process images of any size by a fully convolution network structure. 3. Our method can detect both motion and out of focus blur regions. 4. Our method does not require clear images as references.

However, there are also two disadvantages of the proposed method that need to be improved. One problem is that the segmentation result will be not accurate if the blur boundary is not sharp enough, especially for images with motion blur. Another problem is that when large texture flatten areas appear, the segmentation result may be not satisfactory. We think this problem can be solved by fusing more global context information or training on datasets with larger size images.

In the future work, we will pay more attention to data acquisition and the invention of the more effective network for the challenging blur detection task.

REFERENCES

[1] J. Biemond, *Basic Methods for Image Restoration and Identification*. Amsterdam, The Netherlands: Elsevie, 2009.

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

[3] 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.

[4] R. Liu, Z. Li, and J. Jia, "Image partial blur detection and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[5] A. Chakrabarti, T. Zickler, and W. T. Freeman, "Analyzing spatially-varying blur," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2512–2519.

[6] S. Zhuo and T. Sim, "Defocus map estimation from a single image," *Pattern Recognit.*, vol. 44, no. 9, pp. 1852–1858, Sep. 2011.

[7] X. Zhang, R. Wang, X. Jiang, W. Wang, and W. Gao, "Spatially variant defocus blur map estimation and deblurring from a single image," *J. Vis. Commun. Image Represent.*, vol. 35, pp. 257–264, Feb. 2016.

[8] J. Tiwari, R. K. Rai, and B. Shrman, "A review on estimation of defocus blur from a single image," *Int. J. Comput. Appl.*, vol. 106, pp. 46–49, Jan. 2014.

[9] C. Tang, C. Hou, and Z. Song, "Defocus map estimation from a single image via spectrum contrast," *Opt. Lett.*, vol. 38, no. 10, p. 1706, May 2013.

[10] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 653–664, Mar. 2017.

[11] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 1006–1012, Aug. 2017.

[12] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Comput. Sci.*, to be published.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*, 2015, pp. 234–241.

[19] B. Su, S. Lu, and C. L. Tan, "Blurred image region detection and classification," in *Proc. 19th ACM Int. Conf. Multimedia (MM)*, 2011, pp. 1397–1400.

[20] 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.

[21] G. Liu, S. Chang, and Y. Ma, "Blind image deblurring using spectral properties of convolution operators," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5047–5056, Dec. 2014.

[22] N. Zon, R. Hanocka, and N. Kiryati, "Fast and easy blind deblurring using an inverse filter and PROBE," in *Proc. Int. Conf. Comput. Anal. Images Patterns*. Cham, Switzerland: Springer, 2017, pp. 269–281.

[23] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1033–1041.

[24] Q. Chen, D. Li, and C.-K. Tang, "KNN matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2175–2188, Sep. 2013.

[25] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[26] R. Yan and L. Shao, "Blind image blur estimation via deep learning," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1910–1921, Apr. 2016.

[27] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 769–777.

[28] A. Punnappurath, Y. Balaji, and M. Mohan, "Deep decoupling of defocus and motion blur for dynamic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 750–765.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[30] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: http://arxiv.org/abs/1605.07146

[31] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2016, *arXiv:1611.05431*. [Online]. Available: http://arxiv.org/abs/1611.05431

[32] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[33] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015.

[35] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," *Comput. Sci.*, to be published.

[36] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *Comput. Sci.*, to be published.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[38] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," 2016, *arXiv:1606.02585*. [Online]. Available: http://arxiv.org/abs/1606.02585

[39] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.* vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Apr. 2015.

[41] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning," 2016, *arXiv:1608.08614*. [Online]. Available: https://arxiv.org/abs/1608.08614

[42] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," 2014, *arXiv:1408.5093*. [Online]. Available: http://arxiv.org/abs/1408.5093

[43] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[44] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[45] Y. Kong, Y. Deng, and Q. Dai, "Discriminative clustering and feature selection for brain MRI segmentation," *IEEE Signal Process. Lett.*, vol. 22, no. 5, pp. 573–577, May 2015.

[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, and Z. DeVito, "Automatic differentiation in PyTorch," in *Proc. 29th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–4.

[47] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.

[48] 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, vol. 1, no. 2, pp. 596–605.

[49] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2096–2106, Aug. 2007.

[50] O. H. Maghsoudi, A. Vahedipour, B. Robertson, and A. Spence, "Application of superpixels to segment several landmarks in running rodents," *Pattern Recognit. Image Anal.*, vol. 28, no. 3, pp. 468–482, Sep. 2018.

[51] 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.

[52] B. Kim, H. Son, S. Park, S. Cho, and S. Lee, "Defocus and motion blur detection with deep contextual features," *Comput. Graph. Forum*, vol. 37, no. 7, pp. 277–288, Oct. 2018.

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

AODONG SHEN received the B.S. degree in electrical technology and the M.S. degree in communication and information engineering from Xidian University, Xi'an, China, in 2000 and 2003, respectively, and the Ph.D. degree in biomedical engineering from Southeast University, Nanjing, China, in 2010. From 2003 to 2004, he was an Engineer with the Nanjing Institute of Electronic Technology, CTEC, Nanjing. He is currently a Teacher with the School of Computer Science and Engineering, Southeast University, Nanjing. His current research interests span the areas of image processing, machine learning, and data mining, with an emphasis on the application of deep learning.

HAN DONG is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Southeast University, China. His research interests include image processing and pattern recognition. He is currently working on a machine vision research and developing a machine-vision-based weld seam tracking system for enterprises.

KUN WANG is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Southeast University, China. His research interests include semantic segmentation, object recognition, medical image processing, and pattern recognition. His research interests also include protein structure prediction and computational biology.

YOUYONG KONG received the B.S. and M.S. degrees in computer science and engineering from Southeast University, Nanjing, China, in 2008 and 2011, respectively, and the Ph.D. degree in imaging and diagnostic radiology from The Chinese University of Hong Kong, Hong Kong, in 2014. He is currently an Associate Professor with the College of Computer Science and Engineering, Southeast University. His current research interests include machine learning, and medical image processing and brain network analysis.

JIASONG WU (Member, IEEE) received the B.S. degree in biomedical engineering from the University of South China, Hengyang, China, in 2005, and the joint Ph.D. degree from the Laboratory of Image Science and Technology (LIST), Southeast University, Nanjing, China, and Laboratoire Traitement du signal et de l'Image (LTSI), University of Rennes 1, Rennes, France, in 2012. He is currently working with LIST as a Lecturer. His research interests mainly include deep learning, fast algorithms of digital signal processing and its applications. He received the Eiffel Doctorate Scholarship of Excellence from the French Ministry of Foreign Affairs, in 2009 and the Chinese Government Award for Outstanding Self-Financed Student Abroad from the China Scholarship Council, in 2010.

HUAZHONG SHU (Senior Member, IEEE) received the B.S. degree in applied mathematics from Wuhan University, China, in 1987, and the Ph.D. degree in numerical analysis from the University of Rennes 1, Rennes, France, in 1992. He is currently a Professor with the LIST Laboratory and also the Co-Director of CRIBs. His recent work concentrates on the image analysis, pattern recognition, and fast algorithms of digital signal processing. He is a Senior Member of the IEEE Society.