Deep residual deconvolutional networks for defocus blur detection

Kai Zeng1,2 | Yaonan Wang1,2 | Jianxu Mao1,2 | Xianen Zhou1,2

1 The School of Electrical and Information Engineering, Hunan University, China
2 The National Engineering Laboratory for Robot Visual Perception and Control Technology, Hunan University, China

Correspondence Jianxu Mao, The School of Electrical and Information Engineering, Hunan University, China. Email: maojianxu@hnu.edu.cn

Funding information National Natural Science Foundation of China, Grant/Award Numbers: 61733004, 61573134; National Key Research and Development Project, Grant/Award Number: 2020YFB1712600

Abstract

Accurate defocus blur detection has instigated wide research interest for the last few years. However, it is still a meaningful yet challenging machine vision task, and most methods rely on prior knowledge. Convolutional neural networks have proved the huge success for different tasks within the computer vision, and machine learning flew. A simple yet effective method of defocus blur detection was proposed in this paper, which by applying the deep residual convolutional encoder-decoder network. The aims of DRDN is to automatically generate pixel-level predictions for defocus blur images, and reconstruct output detection results of the same size as the input, which by performing several deconvolution operations at multiple scales through the transposed convolution, and skip connection. Afterwards, we used the slide window detection strategy and traversed the input image with a certain stride. Experiments on challenging benchmarks of defocus blur detection show that our algorithm achieved state-of-the-art performance, and powerfully balanced the detection accuracy, and detection time.

1 | INTRODUCTION

In the field of digital images, there is a very common phenomenon: Defocus blur. It is the result of an out-of-focus optical imaging system [20]. The defocus blur can be used to directly attract the viewer’s attention and emphasize the major subject of a picture by transforming the foreground and/or background blurry. Hence, the estimation of the defocus blur is very important to select the major scene information [40]. Moreover, defocus blur prediction is a significant component in many image processing and machine vision tasks, such as image restoration, photo editing, depth recovery, and image segmentation. Hence, accurately detecting defocus blur is important to the preprocessing or post-processing of machine vision tasks.

The human visual system excels at detecting the localized blur of defocused images, but the underlying mechanism is not well understood. The traditional method of blur detection usually extracted the local image features as local blurrd metric to fit the underlying mechanism of human visual system.

Early works on defocus blur detection put a lot of effort into obtaining the local blurrd metric, which was applied the blurred image features, such as (a) the intensity domain features: Singular value decomposition [4], linear discriminant analysis [2], and sparsity [5]; (b) the gradient domain features: Gradient histogram span [1] and kurtosis [2, 3]; (c) the frequency domain features: Frequency spectrum [7] and power spectrum [4, 6]. Nearly all previous local blur mappers relied on the local blur features which better represent the blur regions. Those local metric methods are effectively detected the image blur, but with limited success, explicitly or implicitly, and failed to accurately discriminate sharpness and blurred regions.

At present, deep learning methods such as deep belief network (DBN) [59], Faster RCNN [60], Yolo [61] etc. has proved a huge success in different tasks of machine vision such as RGB image processing, CT image processing, SAR image processing, and so on, without relying on any prior image or video knowledge. Taking the RGB image processing for instance Zeng et al. [40] used the convolution neural networks (CNN) to successfully learn the locally relevant feature information of defocus images and then automatically estimated the defocus blur images by applying the local metric map.

Currently, the successful experience of CNN was extended to solve the dense prediction problems. Defocus blur detection is similar to the task of dense output prediction, which is where every pixel in the input image are predicted with a probability label, resulting in an estimation results map whose as same as of the input image size.
A usual method to obtaining the dense estimation result is to extract a fixed-sized local map centred on each pixel and then by applying the regular CNN to classify or determine the center pixel label, or to use the fully convolution networks (FCNs) to reconstruct the full-sized input of dense output predictions [41]. The full-resolution, pixel-level output reconstruction also addresses the problem of incorporating sufficient contextual information. The performance of the FCNs method is limited by the FCN operation to reconstruct the prediction results with the same as the input size. There is some feature information within the full-resolution map that is lost and not well preserved.

The aim of this paper is to propose a simple yet powerful model to automatically detect the defocus blur image without relying on prior knowledge. This model used the deep residual deconvolutional autoencoder network to generate the dense output from reconstructing the full-size input of pixel level prediction. We are adding the multi-scale residual connections between every several blocks of the convolution and deconvolution module to enhance the performance of convolutional networks. Experiments on challenging benchmarks of defocus blur detection show that our algorithm achieved state-of-the-art performance and powerfully balanced the detection accuracy and detection time. The more detailed comparison results introduce in Section 4.

2 RELATED WORKS

Accurate defocus blur detection has always been a meaningful and challenging machine vision task. In the past two decades, there have been numerous attempts [1, 2, 4–7, 10, 13, 18–20, 24, 25, 62–69] to the image blur estimation. And the wide research of image blur detection is mainly focused on the local metric method.

Recently, Su et al. [1] propose a local image features hybrid approach that combining four local image blur features together, such as power spectrum, gradient histogram, auto-correlation congruency, and maximum saturation to predicted the sharpness areas and classify blurred or sharpness images. However, this method didn’t know what exact kernels that mainly affected the detection accuracy change. Liu et al. [4] proposed a method that extracts the blurred image local patch to be decomposed by singular value decomposition (SVD). The largest singular values of SVD are simply used as the threshold to segmented the blurred area. Shi et al. [2] proposed a novel method that handles the image blur estimation as a local image classification task. By training the Bayes classifier with the local gradient histogram span and kurtosis features, the local blurred area is to be output as the likelihood distribution and classified as the blur label. Meanwhile, Tang et al. [18] presented a blur metric based on the log averaged spectrum residual to get a coarse blur map. Then, a novel iterative updating mechanism is proposed to refine the blur map from coarse to fine by exploiting the intrinsic relevance of similar neighbouring image regions. Florent et al. [13] treat the blur kernel estimation as a multi-label energy minimization problem by combining learned local blur evidence with global smoothness constraints [11].

Blind deconvolution is an important measure of defocus blur detection. Those methods usually can be divided into two stages: blur kernel estimation [62–64] and non-blind deblurring [65–67]. It is a typical application for blur deblurring by using the Lucy-Richardson algorithm. Based on blur kernel estimation for defocus blur, Cho [63] convergence analysis the maximum a posteriori (MAP) problem of blind deconvolution. Schmidt [69] uses a Bayesian minimum mean squared error estimate (MMSE) instead of a MAP estimate to model the image deblurring. For non-blind image deblurring, Helstrom [65] proposed the Wiener filtering non-blind deblurring algorithm of defocus deblurring, Schuler [67] using a neural network to remove the coloured noise effect in non-blind image deconvolution. Texture features are also important for defocus blur detection. There are four common types of textures that appear in natural scenes, a random texture such as grass, a man-made texture, a smooth texture such as sky or fruit surface, and an almost smooth texture such as areas on the road sign [20, 68]. Yi et al. [20] exploiting those images texture features and discovers an interesting phenomenon that existing the different distribution of uniform local binary patterns between the sharpness areas and blurred areas. This phenomenon was used to segment the sharpness area and the blurred area. Although these methods are effective, the detection performance needs to be further improved.

Some novel algorithms, different from the showing above methods, are exploiting the local frequency or magnitude spectrum to designed the local blur detection metric. Vu et al. [6] presented an algorithm to measure the local perceived sharpness in an image. They have utilized both spectral and spatial properties of the image. For each local patch, this method measures the slope of the magnitude spectrum and the total spatial variation. These measures are then adjusted to account for visual perception, and then, the adjusted measures are combined via a weighted geometric mean. Zhu et al. [7] tried to explicitly estimate the space-variant PSF by analyzing the localized frequency spectrum of the gradient field and takes smoothness and colour edge information into consideration to generate a coherent blur map indicating the amount of blur at each pixel. In [24], Tang et al. use the relationship between the amount of spatially-varying defocus blur and spectrum contrast at edge locations to estimate the blur amount at the edge locations. Then a defocus map is obtained by propagating the blur amount at edge locations over the image using a non-homogeneous optimization procedure. Tang et al. [24] presented an algorithm estimating a defocus scale map from a single image, which is applicable to conventional cameras. This method is capable of measuring the probability of local defocus scale in the continuous domain. It also takes smoothness and colour edge information into consideration to generate a coherent blur map indicating the amount of blur at each pixel. Those methods can effectively detect the defocus blur regions, but the estimation results of those methods still include some incorrectly labelled areas compared with the ground-truth.
Unlike these methods that just only focus on local image features information to construct a local metric for defocus blur detection, Shi et al. [5] propose a simple yet effective blur feature via sparse representation and image decomposition. The sparse dictionary is constructed based on a large external set of defocus images. This method can directly establish the correspondence between sparse edge representation and blur strength estimation.

Another novel method is utilizing the single image depth map estimation for defocus blur regions segmentation. That approach is mainly paying attention to the location of the blurred edge. And then the estimated blur amount is expended over the entire defocus blur image. Zhuo et al. [10] recovers the image depth map and re-blurred the defocus blur image by using a Gaussian kernel. And then the defocus blur amount can be obtained from the ratio between the gradients of input and re-blurred images. By propagating the blur amount at edge locations to the entire image, a full defocus map can be obtained.

Currently, deep learning has proved a huge success in different tasks of machine vision without any prior image or video knowledge. The successful experience of deep convolutional neural networks have begun to permeate through the whole machine learning and machine vision flew, such as the image classification [21][22], segmentation [46] and object detection [23][35] tasks. Ahmed et al [46] proposed the residual deconvolutional networks for brain electron microscopy image segmentation. They took the image segmentation tasks as dense output prediction problems. Alex Kendall [47] proposed deep convolutional encoder networks architecture for the image semantic pixel-wise segmentation. That segmentation engine consisted of an encoder network, and a corresponding decoder network followed by a pixel-wise classification layer.

The excellent performance of CNNs has been extended to blur detection and classification. Zeng et al. [40] used the CNN to successfully learn the locally relevant feature information of defocus images and then automatically estimated the defocus blur images by applying the local metric map. Jinsun Park et al. [44] introduced robust and synergetic handcrafted features and a simple but efficient deep feature from a CNN architecture for defocus estimation. A sparse defocus map is generated using a neural network classifier followed by a probability-joint bilateral filter. The final defocus map is obtained from the sparse defocus map with guidance from an edge-preserving filtered input image. Those methods no longer required the laborious design of local metrics for defocus blur detection. But they just used CNN to extracted the local image blur...
feature for defocus blur detection. Not an end-to-end CNN architecture, will not get a significant improvement in detection performance.

Wang et al. [43] exploiting a fully connected CNN architecture to classify four blur types of images: defocus blur, Gaussian blur, haze blur, and motion blur. The supervised learning model created to map the input images into a higher dimensional feature space, in which the blurs can be classified accurately. Comparison with encoder-decoder architecture, the fully connected layer will lose feature information and can’t obtain better performance.

Accurate defocus blur detection has instigated wide research interest for the last few years. However, it is still a meaningful yet challenging machine vision task. Although the above-mentioned methods can effectively detect the defocus blur regions, but the estimation results of those methods still include some incorrectly labelled areas compared with the ground-truth.

The aim of this paper is to propose a simple yet effective method to automatically detect defocus blur. Neither similar to [40, 44] that just used CNN to extracted the local image blur feature for defocus blur detection, nor similar to [43] that used the fully connected architecture for image blur classify, we proposed end-to-end deep residual convolution encoder-decoder architecture for defocus blur detection. We considered the defocus blur detection as a problem of dense output prediction and reconstructed a full-size input of pixel-level prediction by applying several deconvolution operations at multiple scales through aggregated bilinear interpolation. A more detailed introduction to the method is outlined in Section III.

3 | PROPOSED METHOD

This paper proposed a deep symmetric autoencoder residual network for defocus blur detection, which aimed to generate pixel-level predictions for defocus blur images and reconstruct an as same size as input image by applying several multiple scales deconvolution layer with the transposed convolution and skip connection. We used the sliding window detection strategy and the selective area of local prediction result to detect the raw defocus blurred image. The block diagram of our proposed method shown in Figure 1. An illustration of the DRDN architecture is shown in Figure 2. More detailed information of that method is introduced in Section III.

3.1 | Deep Residual Deconvolutional Network

Recently, the end-to-end deep learning architecture has proved the huge success in the machine learning field. The aim of this paper is to exploit the end-to-end encoder-decoder convolution networks to defocus blur detection, which replaced that just applying CNN extracted the local defocus image features for defocus blur detection. Meanwhile, in comparison with encoder-decoder architecture, the fully connected layer will lose feature information and cannot obtain better performance.

| Table 1 | The drdn configurations |
|----------|-------------------------|
| DRDN configuration | |
| Stage 4 | max-pooling | Transposed convolution | Stage 5 |
| conv3-256(x41) | conv3-256(x51) |
| conv3-256(x44) | conv3-256(x52) |
| Add(x41,x43) | conv3-256(x53) |
| conv3-256(x43) | Add(x51,x53,x41,x44) |
| conv3-256(x42) | conv3-256(x54) |
| Stage 3 | max-pooling | Transposed convolution | Stage 6 |
| conv3-192(x36) | conv3-192(x61) |
| Add(x31,x35) | conv3-192(x62) |
| conv3-192(x35) | conv3-192(x63) |
| conv3-192(x34) | conv3-192(x64) |
| conv3-192(x33) | conv3-192(x65) |
| conv3-192(x32) | Add(x61,x65,x31,x36) |
| conv3-192(x31) | conv3-192(x66) |
| Stage 2 | max-pooling | Transposed convolution | Stage 7 |
| conv3-96(x25) | conv3-96(x71) |
| Add(x11,x13) | conv3-96(x72) |
| conv3-96(x24) | conv3-96(x73) |
| conv3-96(x23) | conv3-96(x74) |
| conv3-96(x22) | Add(x71,x74,x21,x25) |
| conv3-96(x21) | conv3-96(x75) |
| Stage 1 | max-pooling | Transposed convolution | Stage 8 |
| conv3-48(x14) | conv3-48(x81) |
| Add(x11,x13) | conv3-48(x82) |
| conv3-48(x13) | conv3-48(x83) |
| conv3-48(x12) | Add(x81,x83,x11,x14) |
| conv3-48(x11) | conv3-48(x84) |
| Input(96 × 96 × 3) | Output(conv3-1) (96 × 96 × 1) |

The autoencoder (AE) was originally developed for unsupervised feature learning from noisy inputs but is also suitable for image reconstruction. In the context of image denoising, CNN also demonstrated excellent performance. However, with the neural network depth increases, and the image down-sampling operated, the encoder-decoder architecture will appear in gradient disappears and the reconstructed image lose much more detail information. Currently, residual networks [33] demonstrated to solve the above problems and obtained improved performance on tasks of image segmentation and classification. Hence, in combination with the residual network and symmetric convolutional-deconvolutional layers [34], we represented a deep residual deconvolutional autoencoder network to generate dense output prediction for defocus blur detection.

As shown in Figure 2, we presented a fully symmetric convolutional-deconvolutional network. This architecture has better performance for pixel level defocus blur prediction. The convolutional stage is mainly responsible for extracting the feature representations of defocus blur. The deconvolutional
FIGURE 3  The average running times of the different stride size for defocus blur image detection

FIGURE 4  The precision-recall comparison. (a) The comparison of precision-recall curves for different methods with the detection stride size: 10.(b) The comparison of precision-recall curves of the different stride size for defocus blur image detection

FIGURE 5  The comparison of average precision-recall histogram by averaging over all precision and recall levels with varying the threshold within the range [0,255]
stage is mainly responsible for reconstructing the full-size input of the blur or sharpness regions and producing the pixel level defocus blur prediction. We also enhanced the performance of DRDN by adding multiply residual connections between every several blocks of convolution or deconvolution layers.

In this architecture, the local relevant features strengthen paths have been added to the deconvolution and convolution layers. The image context feature maps learned from the convolutional stage were projected and added into the deconvolutional decoding stages to strengthen the performance of generated pixel-level dense probability maps. The context feature maps not only added between the convolution stage and the deconvolution stage, but also added into the convolution block or deconvolution block. We used the max-pooling and deconvolution operator to down-sampling and up-sampling the feature map size and also ensure the symmetric convolution block and deconvolution block with the same feature map size and channels. There was no additional processing when we performed skip-connection before adding the skip paths.

We used a fully convolutional encoder-decoder architecture that took the input of the arbitrary size and reconstructed the full-size input of results by applying the deconvolutional networks. Whenever size preservation was needed, zero padding was always used in the learned layers. The deconvolution layer was implemented using the transposed convolution and fixed (bilinear) filter kernel. Last deconvolution layer of DRDN architecture was used as output by applying one convolutional kernel to correspond the binary ground-truth image labels. The activation function of each convolution or deconvolution layer is activated by rectified linear units (RELU), which were used as the non-linearity transformation. The batch normalization is also used in this network. The detail configuration of DRDN architecture is showing in Table 1.

For the DRDN architecture trained, we used the backpropagation algorithm to deliver the network error computed by the mean-square loss function. Meanwhile, updating the network weight by applying the stochastic gradient descent algorithm. The architecture weight was initialized with randomly Gaussian distribution drawn samples with zero mean and variance. The standard colour augmentation in [70] is used with the per-pixel mean subtracted.

3.2 Defocus blur image dense prediction problem

This paper presented a symmetric encoding and decoding residual network for defocus blur prediction. We extracted the $96 \times 96 \times 3$ local patch maps in training. And the full size image was used in testing. However, there was a problem with image pixel-level dense prediction in that the prediction accuracy of the pixels around the image could not meet the expectations, especially in training the neural network with local patches. This was because local feature information of pixels surrounding the image may have been lost due to the presence of image boundaries. Therefore, neural network models predicting these pixels may not meet our expectations. We can improve the method in two ways: 1) we will do our best to optimize the structure of the neural network model to improve the prediction performance and, 2) we will appropriately add some local patches containing the raw defocus image boundaries into the training database and remove the pixels around the prediction results when predicting.

This paper presented a better performance neural network structure than previously developed. In the process of defocus blur detection, it was time consuming to extract local patches centered on each pixel of the image, and determine the properties of those pixels by neural network model prediction:
blurriness or sharpness. Hence, we used the slide window detection strategy and traversed the input image with a certain stride. However, the size of the input image will not be an integer multiple of the step size in most cases. So we need to extend the input image and traversed the extended image with a certain stride to extract the local patch as an input to the DRDN architecture. There was a phenomenon with this detection strategy. When the sliding step size was smaller than the partial patch size, the adjacent local patches contained a part of the overlapping area, so this part was detected multiple times. How should we deal with this situation? The aim of this paper is to improve the detection accuracy of the blurred image, we only selected a part area centered on each local patch prediction output as the final detection result of this patch. And the detection stride size was equal to the size of the selected area centered on each local patch prediction output. We balanced the relationship between stride size, local patch size and defocus blur image size to better perform defocus image detection.

4 | EXPERIMENTAL RESULTS

4.1 | Implementation details

Our architecture performing on the publicly available machine learning platform: Tensorflow [39]. We trained the DRDN
architecture with Adam optimizer \([38]\) with \(\beta_1 = 0.9, \beta_2 = 0.999\) instead of stochastic gradient descent. The batch size in this paper is 64 and the epoch of training DRDN architecture is 30. The initial learning rate is set to \(r = 0.001\). It is down-scaled by 10 after epoch 20. The experiments were carried out on an NVIDIA RTX 2080 GPU machine, requiring approximately three days of training. We evaluate the performance of our DRDN architecture in this section. The datasets and evaluation metric is described in Section 4.2. The comparison results of our method are discussed in Section 4.3.

4.2 Datasets and evaluation metric

We extract the equal-sized local image map centering the interesting pixel form the defocus image. To obtaining the interesting pixel, we first clustering homogeneous pixels by applying the SLIC method \([36]\) and then extract the gravity centre of each super-pixels area. In other words, the gravity centre pixels are the interesting pixels of our finding. Afterwards, much \(W_p \times W_p\) local patch maps are extracted and with the accompanying ground-truth binary images patch maps. Meanwhile, we appropriately added some local patches containing the raw defocus image boundaries into the training database.

The public defocus blurred image dataset \([50]\) was used by the methods of performance evaluation. There are 704 partially defocus blurred images and the accompanying ground truth binary images in that dataset. We used the introduction above method to construct the training dataset. Meanwhile, we appropriately added some local patches containing the raw defocus image boundaries into the training database. The detection results introduced as following are always gray-scale images and the pixel value within the range \([0, 255]\).

| Methods  | Code       | Avg. Running Times(s) |
|----------|------------|------------------------|
| shi15 [5]| matlab     | 15.76                  |
| Tang [18]| matlab     | 1.78                   |
| Alireza [42]| matlab& cex | 293.59               |
| Zhou [10]| matlab& cex | 13.75                  |
| Zeng [40]| matlab     | 4.8                    |
| Ours     | TensorFlow | 0.27                   |

The public defocus blurred image dataset [50] was used by the methods of performance evaluation. There are 704 partially defocus blurred images and the accompanying ground truth binary images in that dataset. We used the introduction above method to construct the training dataset. Meanwhile, we appropriately added some local patches containing the raw defocus image boundaries into the training database. The detection results introduced as following are always gray-scale images and the pixel value within the range [0, 255].
The evaluation metric of defocus blur detection is evaluated by precision-recall curves, which is generating by varying threshold to segment the final detection results:

\[
\text{precision} = \frac{U \cap U_G}{U}, \quad \text{recall} = \frac{U \cap U_G}{U_G} \quad (1)
\]

where \(U_G\) is a series of the pixel in the ground truth sharpness area and \(U\) corresponding to the pixels within segmented sharpness area from varying threshold segmentation.

The F-measure will also used for evaluating the performance of detection methods.

\[
F_\beta = \frac{(\beta^2 + 1)\hat{P}\hat{R}}{\beta^2\hat{P} + \hat{R}} \quad (2)
\]

where \(\beta\) is the adjustable parameters of F-measure, \(\hat{P}\) denoted the precision, \(\hat{R}\) denoted the recall. We denoted the \(\beta = 0.3\) to evaluated the detection methods in this paper.

### 4.3 Experimental results

The aim of this paper is to propose a symmetric encoding and decoding residual networks for defocus blur detection. The detail configuration information of DRDN architecture is showing in Table 1. To obtain the precision and recall curve, we are first used each gray-scale level threshold within the range \([0, 255]\) to binarized the final detection results. And then calculated the average value of all precision and recall, and plotted the bar graph of average precision and recall by averaging over all precision and recall levels, as showing in Figure 5. According to the Figure 4, we can find that for the majority of threshold, the precision and recall can both achieve comparative performance (\(\text{precision} > 0.95\) and \(\text{recall} > 0.94\)).

In order to determine the selected region size of each local patches, we compared the impact of different selected region size on defocus blur prediction results. The average running times of the different stride size for defocus blur image detection as shown in Figure 3. The smaller the detection stride size, the more time it takes to detect. Meanwhile, we are showing the precision-recall curves in Figure 4(b) to compare the performance of different stride size of defocus blur image estimation.

The comparison detection results are showing in Figures 6, 7, 8, 9, 10, 11, 12, 13, and 14. We just show the DRDN detection results compare with other eight state-of-the-art methods [1, 2, 4, 5, 10, 18, 25, 40, 42]. All of the comparison results are grayscale image, where sharpness area with higher intensities and blurred area with lower corresponding. And we achieved a better detection performance than others. The F-measure evaluation of eight state-of-the-art methods shown in Table 3. We also achieved the better performance.

There are some methods that can effectively detect the defocus blur regions, but the estimation results of those methods still include some incorrectly labelled areas compared with the ground-truth. To make matters worse, there are algorithms dramatically failed to single defocus blur image detection or can’t detect the defocus blur area completely. Moreover, the DRDN method obtains better performance than others. And it can detect the defocus blur occurred in different scenarios as better as possible.

Our method is also efficient. We compare the average running time of the existing methods using all the defocus images on a computer with 3.7 GHz CPU and 16G RAM. The average runtime evaluation compared with other methods are shown in Table 2. For other methods, we utilize the authors’ implementations. The our method is implemented in Tensorflow with NVIDIA RTX 2080 machine. The project is released online and the code can found in this page: https://github.com/zkwalt/DRDN-Defocus-Blur-Detection.

### 5 Conclusion

The aim of this paper was to propose a deep residual deconvolutional network to address the meaningful yet challenging machine vision task: Defocus blur detection. It generates pixel-level predictions for defocus blur images and reconstruct output detection results of the same size as the input, which by performing several deconvolution operations at multiple scales through the transposed convolution and skip connection. We used the slide window detection strategy and traversed the input image with a certain stride. Experiments on challenging benchmarks of defocus blur detection show that our algorithm achieved state-of-the-art performance and powerfully balanced the detection accuracy and detection time.

### ORCID

Kai Zeng https://orcid.org/0000-0002-2745-1253

### REFERENCES

1. Su, B., et al.: Blurred image region detection and classification. International Conference on Multimedia 2011, Scottsdale, AZ, USA, November 28 - December 1397–1400 (2011)
2. Shi, J., et al.: Discriminative blur detection features. Computer Vision and Pattern Recognition 2965–2972 (2014)
55. Long, J., et al.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Machine Intelligence 39(4), 640–651 (2014)
56. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
57. Bahrami, K., et al.: A novel approach for partial blur detection and segmentation. IEEE International Conference Multimedia Expo 1–6 (2013)
58. Chen, L.C., et al.: DeepLab: Semantic image segmentation with Deep Convolutional Nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Machine Intelligence 40(4), 834–848 (2016)
59. Samadi, F., et al.: Change detection in SAR images using deep belief network: A new training approach based on morphological images. IET Image Processing 13(12), 2235–2264 (2019)
60. Ren, S., et al.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems 91–99 (2015)
61. Redmon, J., et al.: You only look once: Unified, real-time object detection. Proc. IEEE Conf. Comput. Vision and Pattern Recognit. 779–788 (2016)
62. Singh Manoj Kumar, Tiwary Uma Shanker, Kim Young-Hoon: An Adaptively Accelerated Lucy-Richardson Method for Image Deblurring. EURASIP Journal on Advances in Signal Processing 2008(1), (2007). https://doi.org/10.1155/2008/365021
63. Cho, S., Lee, S.: Convergence Analysis of MAP Based Blur Kernel Estimation. IEEE Int. Conf. Comput. Vision (ICCV) (2017)
64. Lu, Y., et al.: Blind deblurring for dermoscopy images with spatially-varying defocus blur. IEEE 13th International Conference on Signal Processing (ICSP) 7–12 (2016)
65. Helstrom, C.W.: Image Restoration by the Method of Least Squares. Journal of the Optical Society of America 57(3), 297 (1967)
66. Schmidt, U., et al.: Discriminative non-blind deblurring. Proc. IEEE Conf. Comput. Vision and Pattern Recognit. 604–611 (2013)
67. Schuler, C.J., et al.: A machine learning approach for non-blind image deconvolution. Proc. IEEE Conf. Comput. Vision and Pattern Recognit. 1067–1074 (2013)
68. Cho, T.S.: Motion blur removal from photographs. Ph.D. Dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, (2010)
69. Schmidt, U., et al.: Bayesian deblurring with integrated noise estimation. Proc. IEEE Conf. Comput. Vision and Pattern Recognit. 2625–2632 (2011)
70. Krizhevsky, A., et al.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems 1097–1105 (2012)

How to cite this article: Zeng K, Wang Y, Mao J, Zhou X. Deep residual deconvolutional networks for defocus blur detection. IET Image Process. 2021;15:724–734. https://doi.org/10.1049/ipr2.12057