A Coarse-to-fine Classification for Motion Blur Kernel Size Estimation with Cascaded Neural Networks

Minyuan Ye[1], Lei He[1], Gengsheng Chen[1]*

[1] State Key Laboratory of ASIC and System, Fudan University, No.825 ZhangHeng Rd. Shanghai 201203, China
* Email:gschen@fudan.edu.cn

Abstract
Blur kernel size estimation significantly affects the quality of image deblurring. However, existing methods have a large limitation in estimation accuracy as well as in the adaptability to large kernel sizes. In this paper, we propose a novel blur kernel size estimation method using CNNs. We first convert the 2-D kernel size estimation to a 1-D classification problem. A cascaded network model using homogeneous networks is then designed to cut the entire estimation space into 38 small intervals to have a more accurate kernel size estimation in a coarse-to-fine manner. Experiments show that the proposed method has reached a satisfying performance of kernel size estimation with a largely improved adaptability to various blur kernels.

1. Introduction

A blurry image can be mathematically modeled as a convolution of a latent sharp image and a blur kernel:

$$B = L \otimes k + n \qquad (1)$$

where B, k, L and n denote blurry image, blur kernel, latent sharp image and white noise respectively.

Traditional deblurring algorithms usually use maximum a posteriori (MAP) as their typical framework, where priors are utilized to impose constraints to simplify calculations [1][2][3]. Meanwhile, there are also other methods that use different optimization schemes, such as Bayesian estimation [4] or specific edge prediction [5]. Generally, all these methods have to use blur kernel size as their indispensable input parameter. To get rid of this high dependency, some recent researches turn to use another approach of applying newly-arising machine learning methods to image deblurring, whereas those early neural networks [6][7] still needs a pre-selected range of kernel sizes in order for a satisfying deblurring performance. With more advanced and complicated models be developed, the newest deep learning methods [8][9] do not rely that directly on image's prior information about the blur kernel, but the average PSNR of their restored images (31dB) is still much lower than traditional methods can provide (33dB or even higher), which makes them not applicable to those quality sensitive applications. An accurate estimation of blur kernel size is still a critical problem that affects the image's deblurring quality.

There are already many researches on blur kernel size estimation [10-12]. Liu et al. [10] propose to use two sets of SVMs (support vector machines) to leverage the histogram of oriented gradients (HOGs) at higher levels of an image pyramid. Liu's method is able to fulfill the estimation of blur kernel size but its estimation accuracy still needs to be improved. The same group then proposes another method [11] to use the automap (autocorrelation map of image gradients) for extraction of motion blur information, which largely improves the ability of automap in kernel size estimation. However, its parameters in line detection and recursive filtering have to be set manually according to experience, which makes it difficult to adapt to images with different blur kernels. Li et al. [12], by using the new machine learning method, propose to construct a CNN model to predict the width and the height of the motion trajectories as a regression problem. Li's method can help to obtain a pretty good estimation result when the images' blur kernel sizes are smaller than 35×35, but still has limitation to be applied on large and variable kernel sizes.

In this paper, we propose a new blur kernel size estimation method. With a cascaded structure of homogeneous neural networks, the new method is able to provide with an accurate estimation of the blur kernel size in a coarse-to-fine manner and has a wide-range adaptability to various blur kernels. Our key contributions are as follows:

- We propose to use a patch-selection method to convert the 2-D kernel size estimation to a 1-D classification problem.
- We propose to use a cascaded structure to cut the entire classification space into small intervals to make the estimation more accurate and more adaptable to various blur kernels with different sizes.

2. The proposed method

2.1 Cascaded classification

Since a blur kernel must have its size be an integer, we can consider the kernel size estimation as a classification problem. Moreover, since a blur kernel of a blurry image has most likely its size smaller than 95×95, we further restrict this classification problem to be within the space from 3×3 to 95×95. However, it is still a large range for a quick and efficient classification. So, in this paper, a cascaded classification is introduced to fulfill the blur kernel estimation in a coarse-to-fine manner, in which we divide the entire space into several sub-spaces (or "categories") and then conduct a detailed classification in

978-1-7281-0735-6/19/$31.00 ©2019 IEEE

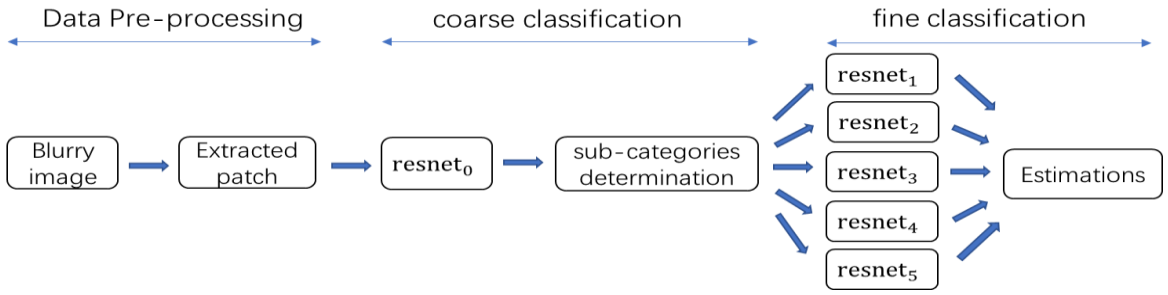

Figure 1. Cascaded classification framework

a smaller sub-space. A cascaded two-stage network is therefore designed.

In the first coarse classification stage, we divide the entire classification space of (3,95] into 5 sub-spaces: (3,19], (19,35], (35,49], (49,63] and (67,95], corresponding to 5 categories of the kernel sizes. In the second fine classification stage, the above 5 sub-spaces are further divided into smaller fine intervals by step size of 2 (4 for the last sub-space) for a detailed address of various kernel sizes.

With such a two-stage structure, we are able to perform an accurate estimation of blur kernel sizes more efficiently by using smaller networks.

2.2 Coarse-to-fine kernel size estimation

The whole estimation contains three steps: data pre-processing, coarse classification and fine classification, as shown in Figure 1.

In data pre-processing, we clip a distinct patch from the blurry image so that all the training data are of the same size. The distinct patch is then selected as follows:

1) We calculate the gradient map Bx and By of each input image in both its x and y directions. Bx and By will be used to estimate the height and the width of the image's blur kernel;
2) In the x direction, with the patch size set to be 128x128, we scan the gradient map Bx patch by patch with a stride of 10 pixels, calculate the standard deviation of each patch, screen out the one with maximum standard deviation and mark it as the distinct patch of this input image. The distinct patch will be used as the input data for training. Note that a patch with large standard deviation indicates its rich retaining of blur kernel information.
3) In the y direction, to simplify the processing and share the network resources, we transpose the blurry image so as to use the same processing as we do in the x direction. In this way, we convert the 2-D classification problem to 1-D classification problem and can use the same network to estimate both the width and the height of the image's blur kernel.

In our cascaded model, we use one residual network ($resnet_0$) to coarsely classify the blur kernel into 5 sub-spaces, and use 5 residual networks ($resnet_{1-5}$) to divide the 5 sub-spaces into 38 small intervals to finely give the detail address of the kernel size, with details shown in Figure 2.

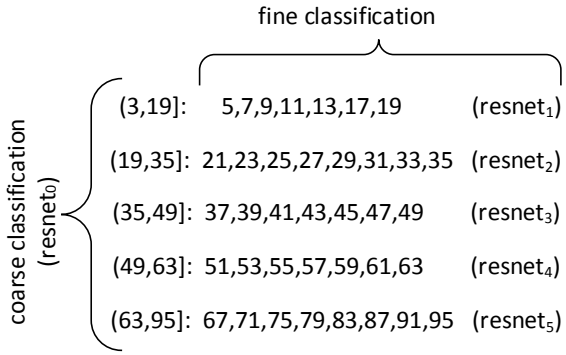

Figure 2. Coarse-to-fine cascaded classification

2.3 Training, validation, and test datasets

The training and validation datasets are generated by convolving blur kernels and sharp images. The blur kernels and sharp images are obtained as follows:

- Sharp images: The MIRFLICKR-25000 image collection contains 24 categories (sky, clouds, water, and etc.). We randomly select 100 images from each category to get our sharp image set $L = \{l_i, i = 1,2,3 \dots 2400\}$;
- Blur kernels: We use the method from [13] to generate all our blur kernels. There are totally 38 different kernel sizes. For each kernel size, we generate 20 different kernels. So the total number of blur kernels is $20 \times 38 = 760$. They form our blur kernel set $K = \{k_i, i = 1,2,3 \dots 760\}$;

A blurry image is then generated by convolving a sharp image with a blur kernel. For each blurry image, we extract a distinct patch to obtain the patch-size pair (p, s). By convolving sharp images in L and kernels in K, we get $2400 \times 760 = 1824000$ patch-size pairs. We use 70% of them for training and 30% of them for validation. The test dataset is used to confirm the estimation results of our model. It is generated by convolving 80 natural images from [14] with 8 simulated blur kernels from [1], which contains 640 blurry images.

2.4 Training details

All the residual networks ($resnet_{0-5}$) used in our cascaded model are of the same structure as the original residual network [15]. Cross-entropy is used as their loss function:

$$\text{Loss} = -\sum_i y_i log(p_i) \qquad (2)$$

where y_i is the ground truth and p_i is the prediction.

In order for an assessment of our cascaded structure, we compare its performance with two other native residual networks (18-layer and 101-layer) using the same loss functions. All the networks are trained with the Adam optimizer and the learning rate is 10^{-4}. Figure 3 depicts the convergence of the 3 networks. From Figure 3 we can see that they all converge quickly in several epochs.

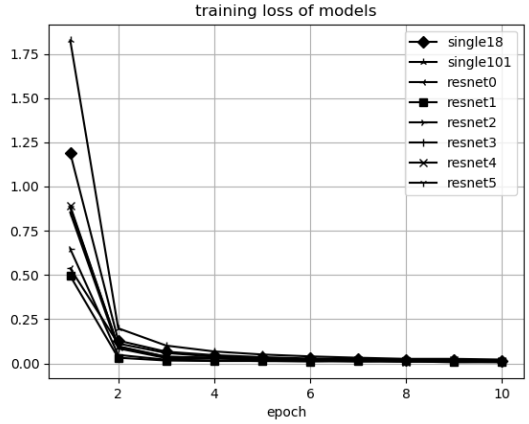

Figure 3. Models converge quickly in several epochs

Table 1. Average absolute error on the test dataset

Type	Native 18-layer	Cascaded 18-layer	Native 101-layer
parameters (each)	1.4×10^7	1.4×10^7	4.3×10^7
parameters (total)	1.4×10^7	8.4×10^7	4.3×10^7
AbE	5.59	4.49	4.04

The average absolute error (AbE) on the test dataset are shown in Table 1. From Table 1 we can see that the cascaded model has its performance of accuracy much higher than the native 18-layer residual network and very close to the 101-layer residual network which is much deeper in network structure. Our cascaded network design is a good balance between the effectiveness and the efficiency. Moreover, because of the use of homogeneous residual network with shallow structure, it is easier to implement the whole cascaded model by hardware.

3. Experiments and analysis

We compare our estimation result on the test dataset with state-of-the-art works [10-12], where [12] gives the best quantitative results. Table 2 presents a comparison of our estimation results with [12]. We can see that our method has obtained a better estimation result (closer in value) against the ground truth (noted as "GT"). And even for the large blur kernels, our method can still have a stable and satisfying estimation performance.

Table 2. Estimation results on test dataset

Kernel	Direction	[12]	Ours	GT[1]
1	Height	21.14	**14.13**	17(15)
	Width	14.98	**8.55**	9(10)
2	Height	20.16	**16.15**	15(16)
	Width	17.32	**11.00**	13(14)
3	Height	17.07	**12.15**	12(11)
	Width	14.29	**12.35**	10(10)
4	Height	**27.19**	15.38	24(24)
	Width	**25.19**	17.30	22(23)
5	Height	**15.15**	7.45	11(12)
	Width	**13.28**	7.05	11(12)
6	Height	**19.15**	12.25	19(20)
	Width	**16.40**	11.95	16(17)
7	Height	**22.00**	11.65	20(22)
	Width	**17.94**	18.15	16(17)
8	Height	24.83	**21.70**	21(21)
	Width	19.68	**16.25**	16(17)
9	Height	-	**65.93**	64
	Width	-	**82.40**	78
10	Height	-	**72.45**	75
	Width	-	**36.15**	39

[1]GT(Ground Truth): the data in the column "GT" in parentheses is provided by [12], which is a little different from the sizes measured by us. We have rechecked the sizes and use the data from its original source [1].

Table 3. PSNRs of the recovered images by method [2]

Kernels	Scene 1		Scene 2		Scene 3		Scene 4	
	O	H	O	H	O	H	O	H
1	31.2	33.8	28.4	28.4	32.6	33.5	30.3	30.5
2	33.0	33.7	28.6	29.9	**34.0**	32.5	**29.1**	28.2
3	**36.2**	34.6	**29.6**	28.5	**35.8**	34.5	**30.9**	29.0
4	32.6	33.1	**28.1**	27.9	32.1	32.2	**30.6**	29.9
5	**30.8**	30.0	22.1	24.4	**34.3**	34.2	**33.8**	33.7
6	27.5	27.7	**22.9**	22.3	25.8	25.8	21.8	22.5
7	28.1	28.2	**24.2**	22.9	**29.9**	29.2	**23.6**	22.9
8	**22.2**	16.0	**16.5**	13.2	**22.2**	17.5	**17.0**	14.7
9	**24.9**	24.7	19.1	20.1	**23.3**	22.8	**21.2**	21.0
10	**24.7**	23.2	17.5	18.1	**23.8**	21.3	**19.6**	17.9
11	**26.1**	24.7	**20.6**	19.7	**25.0**	22.9	**20.4**	19.5
12	25.7	26.8	**20.3**	20.0	**25.4**	24.7	**22.1**	21.8

1. O: Ours 2. H: Human

We then test our model on 48 real blurry images (4 scenes with 12 blur kernels, sizing from 5 to 80) from [16], where the results of most state-of-the-art blind deblurring algorithms are provided by manually adjusting the parameters including the blur kernel sizes. Among them, we select method [2] and [3] for our comparison as they provide executable programs. The restored results are shown in Table 3 and the visualization result is shown in Figure 4. Our model has apparently improved the performance of existing deblurring methods.

4. Summary

Figure 4. Restoration of the blurry image by using deblurring method from [3]. From left to right: blurry input image, deblurring result by tuning kernel size manually [16], deblurring result by obtaining the kernel size using our method.

In this paper, we propose a new blur kernel size estimation method. Throughout the use of a patch-selection method, we convert the 2-D kernel size estimation to a 1-D classification problem. A cascaded network model, using 6 residual networks with the same structure, is then designed to cut the whole estimation space into 38 small intervals and perform the kernel size estimation with a coarse-to-fine manner. Experimental results show that, in comparison with the state-of-the-art works, our new method can provide a better estimation with a more wide-range adaptability to various blur kernel sizes ranging from 3×3 to 95×95, and is able to provide with an efficient and automatic kernel size estimation solution to the existing deblurring algorithms.

References

[1] Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1964-1971. IEEE (2009).

[2] Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-Laplacian priors. In: Advances in neural information processing systems. pp. 1033-1041 (2009).

[3] Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. Acm transactions on graphics (tog) 27(3), 73 (2008).

[4] Wang, C., Sun, L., Cui, P., Zhang, J., Yang, S.: Analyzing image deblurring through three paradigms. IEEE Transactions on Image Processing 21(1), 115-129 (2012).

[5] Cho, S., Lee, S.: Fast motion deblurring. ACM Transactions on graphics (TOG) 28(5), 145 (2009).

[6] Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. IEEE transactions on pattern analysis and machine intelligence 38(7), 1439-1451 (2016).

[7] Hradiš, M., Kotera, J., Zemcık, P., Šroubek, F.: Convolutional neural networks for direct text deblurring. In: Proceedings of BMVC. vol. 10, p. 2 (2015).

[8] Liu, J., Sun, W., Li, M.: Recurrent conditional generative adversarial network for image deblurring. IEEE Access 7, 6186-6193 (2019).

[9] Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multiscale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3883-3891 (2017).

[10] Liu, S., Wang, H., Wang, J., Pan, C.: Blur-kernel bound estimation from pyramid statistics. IEEE Transactions on Circuits and Systems for Video Technology 26(5), 1012-1016 (2016).

[11] Liu, S., Wang, H., Wang, J., Cho, S., Pan, C.: Automatic blur-kernel-size estimation for motion deblurring. The visual computer 31(5), 733-746 (2015).

[12] Li, L., Sang, N., Yan, L., Gao, C.: Motion-blur kernel size estimation via learning a convolutional neural network. Pattern Recognition Letters (2017).

[13] Boracchi, G., Foi, A., et al.: Modeling the performance of image restoration from motion blur. IEEE Trans. Image Processing 21(8), 3502-3517 (2012).

[14] Sun, L., Hays, J.: Super-resolution from internet-scale scene matching. In: 2012 IEEE International Conference on Computational Photography (ICCP). pp. 1-12. IEEE (2012).

[15] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770-778 (2016).

[16] Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: European conference on computer vision. pp. 27-40. Springer (2012)