KERNEL ESTIMATION FOR MOTION BLUR REMOVAL USING DEEP CONVOLUTIONAL NEURAL NETWORK

Yanan Lu[a,b], Fengying Xie[a], Zhiguo Jiang[a]

a: Image Processing Center, School of Astronautics, Beihang University, Beijing, China.
b: Institute of Software, Chinese Academy of Sciences, Beijing, China.
luyanan@iscas.ac.cn, xfy_73@buaa.edu.cn, jiangzg@buaa.edu.cn.

ABSTRACT

Blind deblurring can restore the sharp image from the blur version when the blur kernel is unknown, which is a challenging task. Kernel estimation is crucial for blind deblurring. In this paper, a novel blur kernel estimation method based on regression model is proposed for motion blur. The motion blur features are firstly mined through convolutional neural network (CNN), and then mapped to motion length and orientation by support vector regression (SVR). Experiments show that the proposed model, namely CNNSVR, can give more accurate kernel estimation and generate better deblurring result compared with other state-of-the-art algorithms.

Index Terms— Blind deblurring, motion blur, kernel estimation, deep learning, CNN.

1. INTRODUCTION

As image acquisition equipment become more and more popular, such as digital single lens reflex (DSLR), digital still camera (DSC), and mobile phone, billions of digital photographs are captured every year [1]. Many images come out blurry due to the camera shake, object motion or out-of-focus. Image deblurring aims to restore the sharp image from the blurry version, which has long been an important research topic in computer vision.

Over the past few decades, lots of researchers have been focusing on this topic and developing many remarkable algorithms [2-8]. The vast majority of blind deblurring algorithms can be roughly divided into two groups: 1) algorithms developed based on some priors without learning; 2) learning-based algorithms. Many priors have been utilized to blind deblurring [9-12]. In [10], Cai et al. exploited the prior that the blur kernel is sparse in the curvelet domain and the sharp image is sparse in framelet domain. In [12], Michaeli et al. proposed an algorithm based on the cross-scale patch recurrence property. The main idea of these algorithms above is to build an objective function based on the prior, and then estimate the blur kernels and restore the latent sharp image through maximize/minimize the objective function. Machine learning can extract statistical information from training data and then make predictions for unseen data [13]. Researchers have also applied machine learning to blind deblurring. In [14], Devy et al. learned a logistic regression model using the features extracted by Gabor filters to predict the blur kernel. In [15], Schuler et al. learned a stacked neural network to simulate the iterations in traditional blind deblurring. In [16], Sun et al. transformed the blur kernel estimation to classification problem and used the convolutional neural network (CNN) to predict the image blur kernel.

The blur kernel is crucial for blind deblurring [17]. In this paper, we focus on motion blur kernel estimation. The motion blur kernel can be determined by two parameters: length and orientation. In [16], Sun et al. generated a set of candidate motion kernels by discretizing the length and orientation of the motion kernel and select the most possible one for the input blur image through classification. However, it is unreasonable to mandatorily discrete the continuous motion kernel space to a certain kernel set, which make the model incapable for the motion blur kernels excluded by this set. In this paper, we propose to form a regression model by CNN to predict the length and orientation of the motion blur kernels. In this way, the advantages of CNN can be utilized, and at the same time, the continuity of the motion space can be kept. Through deblurring experiments, the proposed kernel estimation method is verified to be effective for motion blur images. The following of the paper is organized as follows: Section 2 describes kernel estimation. Section 3 introduce a patch selection strategy for better deblurring. Experiments and analysis are presented in section 4. Section 5 concludes the paper.

2. KERNEL ESTIMATION

Given a blur image, the motion blur kernel can be represented using a motion vector $v = (l, o)$, where l and o represent the length and orientation of the motion kernel respectively. In [14], Devy et al. using Gabor features to train a regression model for kernel estimation. While, Gabor feature is hand-

This work was supported by the National Natural Science Foundation of China under grants 61471016 and 61371134. Fengying Xie is the corresponding author, xfy_73@buaa.edu.cn

crafted, which characterizes the responses of certain filter. For some instances, the predicted length and orientation are not accurate enough. Deep learning has been used in many domains, such as speech recognition [18], natural language processing [19], image classification [20], and image super-resolution [21], which can improve the performance dramatically[22]. As a typical deep learning method, the end-to-end CNN model can incorporate feature learning into the training process. In this paper, CNN is used for more accurate kernel estimation, which takes the raw image as input and the motion vector as output.

2.1. Learning a CNN for kernel estimation

In [16], Sun et al. estimated motion kernel through classification based on CNN. However, using limited motion blur kernels to represent all the motion kernels is unreasonable. Here, we train a CNN model to directly predict the motion vector.

Motivted by Sun et al., we use a similar CNN structure as [16]. The designed CNN structure has six layers, as shown in Fig.1. C1 and C3 are convolutional layers. The parameters (i.e. number, size, stride) of filters used are $(96, 7 \times 7, 2)$, $(256, 5 \times 5, 2)$ respectively. Each of the convolutional layer is followed by a non-linear transformation ReLU (i.e., $f(x) = max(x, 0)$). P2 and P4 are pooling layers over 2×2 cells with stride 2. F5 and F6 are fully connected layers with 1024 and 2 outputs respectively. The loss function and optimization method used here is Euclidean distance and Stochastic gradient descentSGD), respectively. Since the blur in different color channels are the same, as many deblurring methods, we translate the input color image to gray scale for blur kernel estimation to reduce the computation complexity. The input of this CNN model should be of fixed size. Considering image resizing will influence the blur extent [12], we use image patches with size of 100×100 as input. The final 2 outputs corresponding to length and orientation of the motion blur kernel respectively. This CNN model is trained and tested using Caffe[1].

In the CNN model, the 1024 outputs of F5 are the deep features extracted from blur images. Treating the connection weights of F6 as the fitting coefficients, the final 2 outputs can be regarded as linear combinations of the deep features.

2.2. CNNSVR

In the CNN model, the deep features are mapped to motion length and orientation by linear regression. Compared with simple linear combination, SVR [23] performs better in handling high dimension regression problem. Therefore, we use the SVR model to replace the final layer in CNN model described in Fig. 1. The combination of CNN and SVR can make them complement each other, and the final estimation accuracy can be further improved. We use the deep features of CNN as the input of SVR (with Radial Basis Function kernel). The length and orientation of motion blur kernels can be calculated as:

$$l = SVR_{CNN\_l}(f_{CNN}) \quad (1)$$

$$o = SVR_{CNN\_o}(f_{CNN}) \quad (2)$$

where, SVR_{CNN_l} and SVR_{CNN_o} are SVR prediction models trained on the deep features f_{CNN}.

[1] http://caffe.berkeleyvision.org/

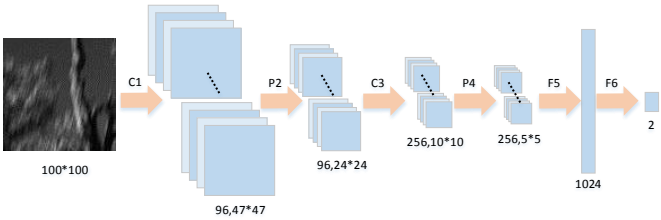

Fig. 1. Structure of the CNN.

3. PATCH SELECTION FOR BLIND DEBLURRING

As the result of visual masking [24], some certain regions of an image can hide blur better than other regions. Figure 2 illustrates the effect of visual masking. Figure 2(a) is a sharp image, and Fig. 2(d) is the blur version of (a), which is obtained by filtering the sharp image using a certain blur kernel. The sky region in the red box and building region in the green box suffered from the same blur distortion. However, we could hardly perceive the blur in the sky when the building is seriously blurred. Because that the blur is obvious in edge regions, while obscure in smooth regions. The smooth regions are easily lead to a wrong kernel estimation result. Therefore, in this paper, we propose a patch selection strategy to choose the patches of an image for blind deblurring.

Fig. 2. Visual masking for image blur.

We define the ratio of the edge pixels for an image and a patch respectively:

$$r_i = \frac{n_i}{N_i} \tag{3}$$

$$r_p = \frac{n_p}{N_p} \tag{4}$$

where N_i represent the pixel number of the image and n_i is the edge pixel number in the image, N_p and n_p represent the pixel number of the image patch and the edge pixel number in the patch respectively. The edge pixels of the image are extracted using Canny operator [25] here. For a patch in an image, if $r_p > r_i$, this patch contains more edge pixels and is selected.

In training stage, the selected patches are used to train the designed network model, and in test stage, the selected patches are used to predict the blur kernel of the corresponding images.

4. EXPERIMENTS AND ANALYSIS

In this paper, a CNN model is designed to learn the deep features from motion blur images, and SVR method is used to map these deep features to the length and orientation of motion blur kernels. The proposed model, namely CNNSVR, uses small image patch as input to estimate the blur kernel. The image patches of an input image are selected using the strategy described in section 3, and then the average estimation result is used to generate the final blur kernel of the input image. With the predicted kernel, the sharp image can be obtained by applying the non-blind deconvolution method described in [26].

In order to quantitatively validate the performance of the proposed algorithm, experiments were conducted in regards to three aspects using our dataset: 1) the effectiveness of patch selection; 2) the kernel estimation accuracy; 3) the deblurring effectiveness.

4.1. Dataset

The CNN model relies on a large amount of training data, so we established a large synthetically blurred image set. We use the 289 sharp images collected in [4]. The scene content of these images varies from indoor to outdoor, from natural scene to man-made environment, from people to animal, and are all from online sources, including Flickr, Facebook, and Google Plus. These images are synthetically blurred using different motion blur kernels. The motion kernels have 8 different length varying from 3 to 24 with interval of 3 and 12 orientations varying from 0° to 165° with interval of 15°. There are 96 ($8 \times 12 = 96$) kernels in total. We synthetically blur the 289 sharp images using the 96 different kernels, and add 0.01 additive Gaussian noise to each blurred image to simulate the sensor noise. Totally, $289 \times 96 = 27744$ blur images are obtained. Of these simulated blur images, the images generated from 279 sharp images are used for training, and the images generated from the remaining 10 sharp images are used for test. Therefore, there are $279 \times 96 = 26783$ images in training set and $10 \times 90 = 960$ images in test set.

4.2. Effectiveness of Patch Selection

We collected two sets of image patches from training images. One set was extracted using the patch selection strategy described in section III, and another set was obtained through randomly selection from the training images. In each set, there are 50000 image patches. We trained the CNN model using the two sets of image patches respectively, and using the two trained CNN models, the blur kernels of the test patches were predicted. Mean absolute error (MAE) between the estimated kernel and the ground truth kernel was used to evaluate the prediction performance of the two trained CNN models, which is defined as:

$$MAE(k, k_g) = \frac{1}{|k|} \sum_{p \in k} |k(p) - k_g(p)| \tag{5}$$

where k and k_g denote the estimated kernel and ground truth kernel respectively. $|k|$ is the number of elements in the kernel k. In our experiments, we use kernels of size 27×27, therefore $|k| = 729$. The lower the MAE value, the better the prediction accuracy of the model. Figure 3 shows the statistics results for 5000 image patches from test images. It can be seen that, the proposed patch selection strategy can remarkably improve the kernel estimation accuracy.

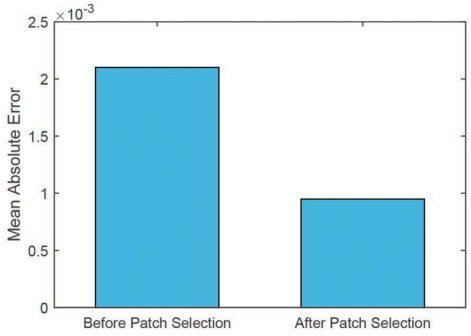

Fig. 3. Effectiveness of patch selection.

4.3. Kernel Estimation Accuracy

For a deblurring method, the correct estimation of blur kernel is crucial to the final deblurring result. We compare the kernel estimation accuracy of the proposed CNNSVR with other state-of-the-art algorithms: EMLO [6], DF [7], RIGP [8], CNN_C[16], and the estimation results of CNN and another regression model GaborSVR, which use Gabor features and SVR to predict the motion kernel, are also used for comparison. For EMLO [6], DF [7], RIGP [8], CNN_C [16], we use the author provided implementation.

Table 1 shows the average MAE on the testing set. It exhibits two points: 1) Among the seven algorithms, CNN_C, CNN and CNNSVR obtain smaller MAE, which means deep learning is effective in kernel estimation. Further, our regression models CNN and CNNSVR performs better than CNN_C, which is because that CNN_C treats kernel estimation as a classification task and its generalization ability is limited; 2) For regression models, the performance is improved significantly when using CNN instead of Gabor features to estimate the motion blur kernel. While, the MAE almost do not change when introduce SVR to the CNN model.

We also use the max absolute error (MaxAE) to further measure the kernel estimation accuracy. MaxAE can be calculated as:

$$MaxAE(k, k_g) = max|k(p) - k_g(p)| \tag{6}$$

where k and k_g are the estimated kernel and ground truth kernel, p is the kernel element. A small MaxAE value indicates a better performance. The mean MaxAE of all the test images are shown in Table 1. It can be seen that the proposed CNNSVR model has the least MaxAE value. Therefore, our CNNSVR model achieves more accurate kernel estimation than all the competing algorithms.

Table 1. Statistical results of kernel estimation and deblurring

Methods	MAE	MaxAE	PSNR	SSIM
EMLO[6]	0.0024	0.1076	19.3357	0.5593
DF[7]	0.0019	0.0897	20.3354	0.6220
RIGP[8]	0.0013	0.0761	21.6111	0.6695
CNN_C[16]	0.0013	0.0734	21.8809	0.6629
GaborSVR	0.0016	0.0914	19.7438	0.5951
CNN	0.0007	0.0594	22.6631	0.6929
CNNSVR	**0.0007**	**0.0538**	**23.0894**	**0.7047**

4.4. Deblurring Effectiveness

The final goal of kernel estimation is to recover the sharp image. We compare the visual quality of the deconvolution result on different kernels estimated by the seven methods. To make a fair comparison, the same non-blind deblurring method is adopted here. Figures 4 and 5 show two examples of the deblurring results. The motion vectors of blur kernel used in Figs. 4 and 5 are (6, 30) and (15,90) respectively. It can be seen that, the results of EMLO, DF, RIGP and CNN_C contain lots of noises. While, the result of GaborSVR is still blur. CNN has almost the same recover result with our CNNSVR method in Fig. 5, while in Fig. 4, the former contains obvious noise compared with the later. Therefore, among these seven kernel estimation methods, our CNNSVR method can generate the most satisfactory deblurring results.

For quantitative comparison, we calculate PSNR(Peak Signal to Noise Ratio) and SSIM (Structure Similarity)[27], two quality metrics, to directly measure the visual difference between the deconvolution results and the corresponding ground truth images (the original sharp images). Larger values of PSNR and SSIM indicate a better restoration result. The statistical results of PSNR and SSIM on 960 test images are shown in Table 1. Obviously, the result of CNNSVR are the best.

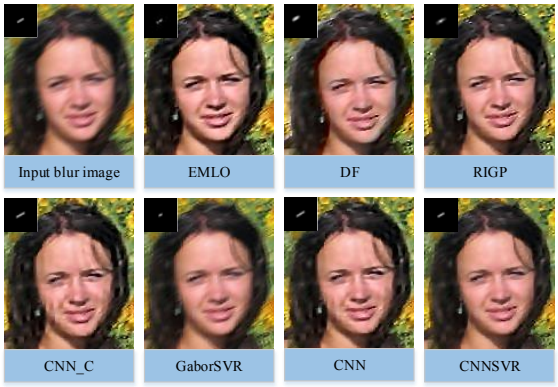

Fig. 4. Deblurring result comparisons for small motion scale.

Fig. 5. Deblurring result comparisons for large motion scale.

5. CONCLUSION

In this paper, a novel motion kernel estimation method CNNSVR is proposed. The edge patches of input image are firstly selected by a strategy for more accurate kernel estimation. Features are then extracted from these patches through the designed CNN model, and mapped to length and orientation of the motion blur kernel by SVR. The proposed CNNSVR utilize both the advantages of deep learning and support vector regression, which can give accurate prediction. Experiment results show that, compared with other state-of-the-art algorithms, the proposed CNNSVR model can give the most accurate kernel estimation and thus achieve the best image deblurring results.

6. REFERENCES

[1] Mittal A, Soundararajan R, Bovik A C. Making a completely blind image quality analyzer. IEEE Signal Processing Letters 20(3), 209-212 (2013)

[2] Chan T F, Wong C K. Total variation blind deconvolution. IEEE Transactions on Image Processing 7(3), 370-375 (1998)

[3] Ayers G R, Dainty J C. Iterative blind deconvolution method and its applications. Optics letters 13(7), 547-549 (1988)

[4] Mai L, Liu F. Kernel fusion for better image deblurring. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 371-380 (2015)

[5] Goldstein A, Fattal R. Blur-kernel estimation from spectral irregularities. In: European Conference on Computer Vision (ECCV). pp. 622-635 (2012)

[6] Levin A, Weiss Y, Durand F, et al. Efficient marginal likelihood optimization in blind deconvolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2657-2664 (2011)

[7] Zhong L, Cho S, Metaxas D, et al. Handling noise in single image deblurring using directional filters. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 612-619 (2013)

[8] Pan J, Hu Z, Su Z, et al. Deblurring text images via L0-regularized intensity and gradient prior. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2901-2908 (2014)

[9] Fergus R, Singh B, Hertzmann A, et al. Removing camera shake from a single photograph. ACM Transactions on Graphics (TOG) 25(3), 787-794 (2006)

[10] Cai J F, Ji H, Liu C, et al. Blind motion deblurring from a single image using sparse approximation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 104-111 (2009)

[11] Cho T S, Paris S, Horn B K P, et al. Blur kernel estimation using the radon transform. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 241-248 (2011)

[12] Michaeli T, Irani M. Blind deblurring using internal patch recurrence. In: European Conference on Computer Vision (ECCV). pp. 783-798 (2014)

[13] Freitag D. Machine learning for information extraction in informal domains. Machine learning 39(2-3), 169-202 (2000)

[14] Couzinie-Devy F, Sun J, Alahari K, et al. Learning to estimate and remove non-uniform image blur. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1075-1082 (2013)

[15] Schuler C J, Hirsch M, Harmeling S, et al. Learning to deblur. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(7), 1439-1351 (2016)

[16] Sun J, Cao W, Xu Z, et al. Learning a convolutional neural network for non-uniform motion blur removal. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 769-777 (2015)

[17] Levin A, Weiss Y, Durand F, et al. Understanding blind deconvolution algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(12), 2354-2367 (2011)

[18] Dahl G E, Yu D, Deng L, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing 20(1), 30-42 (2012)

[19] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: ACM International Conference on Machine learning. pp. 160-167 (2008)

[20] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097-1105 (2012)

[21] Dong C, Loy C C, He K, et al. Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision (ECCV). pp. 184-199 (2014)

[22] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 521(7553), 436-444 (2015)

[23] Smola A J, Schölkopf B. A tutorial on support vector regression. Statistics and computing 14(3): 199-222 (2004)

[24] Legge G E, Foley J M. Contrast masking in human vision. JOSA 70(12), 1458-1471 (1980)

[25] Canny J. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679-698 (1986)

[26] Krishnan D, Fergus R. Fast image deconvolution using hyper-Laplacian priors. In: Advances in Neural Information Processing Systems. pp. 1033-1041 (2009)

[27] Wang Z, Bovik A C, Sheikh H R, et al. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600-612 (2004)