NON-BLIND IMAGE DECONVOLUTION USING DEEP DUAL-PATHWAY RECTIFIER NEURAL NETWORK

Keting Zhang, Weichen Xue and Liqing Zhang*

Department of Computer Science and Engineering, Shanghai Jiao Tong University, China zzsnail82@gmail.com, xueweuchen@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

ABSTRACT

Recently deep neural networks have been successfully used for natural image deconvolution. Whereas the existing methods usually involve an inversion of the blur followed by a denoising step. In this paper we propose a pure learning approach to learn a mapping from a blurred patch to a clean patch directly with a deep dual-pathway rectifier neural network. The experimental results show that our approach outperform the state-of-the-art methods on non-blind image deconvolution within reasonable training time. By analyzing the learned representations, we empirically show that our model works by efficiently detecting the blurry input patterns and then reconstructing the clean patch with the corresponding dictionary atoms.

Index Terms— Non-blind image deconvolution, deep neural network, dual-pathway architecture, rectifier activation function

1. INTRODUCTION

Digital images are often degraded when they are captured due to the shake of the camera and the noise corruption. Image deconvolution aims to recover the clean original image from its degraded observation. Various effective algorithms have been proposed, such as [1, 2, 3, 4, 5], and much progress in performance has been made. These methods are usually based on natural image priors and are well-engineered. Could we learn a pure deconvolution procedure to approximate the mapping from a blurred image to a clean image?

Recently deep neural networks have been used to learn a denoising function from a noisy patch to a clean patch and many network models have been proposed, including stacked sparse autoencoder [6], convolutional neural network [7] and plain neural network [8]. A multi-layer enormous neural network has been shown to achieve state-of-the-art denoising performance [8].

The successful applications of deep neural network to image denoising inspire researchers to study its use in image deconvolution. However, directly applying the typical network models mentioned above to learn a deconvolution function was shown not to be able to yield decent performances [9, 10]. The main reason is what has been learned by these models is still blurry and the sharp information is not preserved.

In this paper, we propose a patch-based pure learning approach for non-blind image deconvolution with a deep dualpathway rectifier neural network (DRNN) [11]. Compared to the conventional neural networks, the DRNN model has been shown to improve the efficiency of capturing information from the noisy data. We approach image deconvolution problem by learning the blurry patterns and the corresponding dictionary atoms. We describe how to adapt DRNN model to learn a deconvolution mapping from a blurred patch to a clean patch. The experimental results for motion deblurring show that our approach outperforms the current state-of-theart methods on some standard test images both quantitatively and visually, while the training time is reasonable. In addition, we empirically demonstrate that our model can detect the blurry input patterns and the corresponding clean dictionary efficiently, and thus succeeds in recovering the blurred images. We provide a Matlab toolbox with the trained models to test our method¹.

2. RELATED WORK

Sparse image priors are used in many image deconvolution methods. Studies of natural image statistics have shown that image gradients follow the heavy-tailed distributions [12]. Various approximations including hyper-Laplacian priors [1, 2, 3] as well as Gaussian Mixture priors [4] have been successfully applied. While Schmidt *et al.* [5] learn a cascade of priors model from the image data.

Another line of research is to utilize deep neural network. Because the typical network models do not work well for image deconvolution [9, 10], the methods based on deep neural network usually take a two-step procedure. Xu *et al.* [10] proposed a deep convolutional network structure consisting of two submodules corresponding to image deconvolution and artifact removal. The first submodule uses separable kernel

^{*}Corresponding author. This work was supported by the Key Basic Research Program of Shanghai under Grant 15JC1400103 and the National Natural Science Foundation of China under Grants 91420302 and 61272251.

¹https://github.com/zzsnail/image-deconvolution-DRNN



Fig. 1. Patch-based image deconvolution using dual-pathway rectifier neural network. (a) our model takes a blurred patch as input and tries to recover its central block as output. (b) a dual-pathway rectifier network with three hidden layers. For every neuron in hidden layers, there is an extra companion node with opposite input and output weights.

inversion as the weight initialization. Schuler *et al.* [9] used a inversion of the blur in Fourier domain as the first step and then removed the colored noise by a plain feed-forward neural network. In addition, Sun *et al.* [13] used convolutional neural network to estimate the motion blur field and then removed the non-uniform motion blur with Gaussian Mixture priors.

Differences to our work: We address the uniform deconvolution problem by detecting the blurry input patterns from the degraded image and learning the corresponding clean dictionary atoms to reconstruct the recovered image. Our model is a pure learning approach directly using deep dual-pathway rectifier network which is essentially a plain feed-forward neural network.

3. OUR APPROACH

Mathematically, we consider the following image degradation model:

$$\tilde{X} = K * X + n \tag{1}$$

where X represents the original clean image and \dot{X} the degraded image. The notation K is the known convolution kernel, or referred to as a point spread function, and n models additive white Gaussian noise.

3.1. Dual-pathway rectifier neural network

Dual-pathway rectifier neural network (DRNN) is one model we proposed recently for image denoising [11]. The motivation is to improve the efficiency of capturing information, which is hurt by rectifier's one-sided property.

A deep DRNN with three hidden layers is shown in Fig. 1(b). DRNN is obtained based on the plain feed-forward rectifier neural network by introducing some extra hidden nodes. For every rectifier neuron in hidden layers, one extra companion node with the opposite input and output weights is added. In Fig. 1(b) we denote the original (upper) and companion (lower) nodes in the *i*-th hidden layer by h_i^u and h_i^l , respectively. First we add the nodes h_1^l and associate weight $-W_1$ with their connection to input layer and weight $-W_2$ with connection to h_2^u . Then the nodes h_2^l are added with associated weight $-W_2$ to h_1^u and $-W_3$ to h_3^u . Thus the weight associated with the connection between h_1^l and h_2^l is W_2 . In this way we get the entire DRNN model.

As shown in [11], the DRNN model is equivalent to a feed-forward neural network with a novel activation function g(x) defined by:

$$g(x) = max(0, x+t) - max(0, -x+t)$$
(2)

where t is the parameter which can be learned simultaneously with all weights and biases. Practically we use the minibatch Limited memory BFGS (L-BFGS) method which was shown to be able to significantly simplify and speed up training deep models [14] to optimize all network parameters.

3.2. Adapting DRNN for patch deconvolution

We want to learn a patch-based deconvolution mapping with DRNN. In equation (1), we make the following sparse decomposition of blurred images:

$$\tilde{X} = K * \left(\sum_{i} a_{i} \phi_{i}\right) + n = \sum_{i} a_{i} \left(K * \phi_{i}\right) + n \qquad (3)$$

where ϕ_i is spatial basis function in clean image (patch) space with coefficient a_i which typically has a sparse distribution [15]. The second step is based on the distributive and associative properties of the convolution operation. This equation tells us a blurred image (patch) can also has a sparse representation. Compared to clean images (patches), the difference is that the atoms in this representation space are blurry and can be generated by convolving the clean atoms with the same blur kernel.

In addition, the regions of blurry atoms $K * \phi_i$ are larger than clean ones ϕ_i , because the information spreads out to larger area due to the convolution operation. Thus, as shown



Fig. 2. Visual comparisons of cropped deconvolution results. Two blur kernels provided by [2] are shown in bottom right of the blurred images and the Gaussian noise level is $\sigma = 5$.

in Fig. 1(a), given a blurred patch \tilde{x} of size $\tilde{d} \times \tilde{d}$ as input, we try to recover its central small block of size $d \times d$. Both sizes should satisfy the following condition:

$$\tilde{d} \ge d + k - 1 \tag{4}$$

where $k \times k$ is the size of blur kernel K. Practically the size \tilde{d} can be a little bigger to obtain slight performance gain.

Thus, given many large blurred patches and the corresponding small clean patches for training, we try to use deep DRNN to learn the blurry atoms and the corresponding clean atoms from them, respectively. And further, we learn a patch deconvolution function.

3.3. Application to non-blind image deconvolution

Our approach consists of both training and testing phases. In training phase, given a blur kernel K, we choose some natural images and blur them to get the degraded images. Then we randomly sample some blurred patches and the corresponding clean patches whose sizes satisfy equation (4) to train a deep DRNN model.

During the testing phase, given a blurred image, we first pad it with mirror reflections of itself across the borders to handle the boundary problem. Then we decompose it into a number of overlapping blurred patches and feed them into the trained DRNN to get the recovered patches. Finally, all recovered patches are put at the corresponding positions and averaged on the overlapping regions via Gaussian weighting.

4. EXPERIMENTAL STUDY

First we empirically evaluate our approach for non-blind motion deblurring. Then we discuss how our approach works by analyzing the trained model.

Table 1. Quantitative comparisons on standard test images by PSNR (dB) for kernel 1 (top) and kernel 2 (bottom).

image	Krishnan	Cho	EPLL	Schmidt	Ours
C.man	26.79	27.27	27.19	26.98	27.73
Barbara	25.72	25.60	26.68	25.80	27.23
Bridge	24.97	25.12	24.88	25.21	25.64
Couple	27.54	27.96	28.11	27.90	28.85
Hill	28.68	28.98	28.97	28.96	29.60
House	30.11	30.82	31.28	30.69	32.07
Lena	30.26	30.61	31.37	30.50	31.84
Man	28.49	28.85	28.86	28.79	29.50
Average	27.82	28.15	28.42	28.10	29.06
image	Krishnan	Cho	EPLL	Schmidt	Ours
image C.man	Krishnan 25.95	Cho 26.54	EPLL 26.07	Schmidt 26.32	Ours 26.92
image C.man Barbara	Krishnan 25.95 24.61	Cho 26.54 24.51	EPLL 26.07 25.02	Schmidt 26.32 24.65	Ours 26.92 25.48
image C.man Barbara Bridge	Krishnan 25.95 24.61 24.27	Cho 26.54 24.51 24.66	EPLL 26.07 25.02 24.23	Schmidt 26.32 24.65 24.58	Ours 26.92 25.48 24.74
image C.man Barbara Bridge Couple	Krishnan 25.95 24.61 24.27 26.98	Cho 26.54 24.51 24.66 27.39	EPLL 26.07 25.02 24.23 27.34	Schmidt 26.32 24.65 24.58 27.36	Ours 26.92 25.48 24.74 27.92
image C.man Barbara Bridge Couple Hill	Krishnan 25.95 24.61 24.27 26.98 28.30	Cho 26.54 24.51 24.66 27.39 28.60	EPLL 26.07 25.02 24.23 27.34 28.54	Schmidt 26.32 24.65 24.58 27.36 28.58	Ours 26.92 25.48 24.74 27.92 28.89
image C.man Barbara Bridge Couple Hill House	Krishnan 25.95 24.61 24.27 26.98 28.30 29.36	Cho 26.54 24.51 24.66 27.39 28.60 30.48	EPLL 26.07 25.02 24.23 27.34 28.54 30.39	Schmidt 26.32 24.65 24.58 27.36 28.58 30.06	Ours 26.92 25.48 24.74 27.92 28.89 31.09
image C.man Barbara Bridge Couple Hill House Lena	Krishnan 25.95 24.61 24.27 26.98 28.30 29.36 29.98	Cho 26.54 24.51 24.66 27.39 28.60 30.48 30.33	EPLL 26.07 25.02 24.23 27.34 28.54 30.39 30.97	Schmidt 26.32 24.65 24.58 27.36 28.58 30.06 30.17	Ours 26.92 25.48 24.74 27.92 28.89 31.09 31.39
image C.man Barbara Bridge Couple Hill House Lena Man	Krishnan 25.95 24.61 24.27 26.98 28.30 29.36 29.98 27.82	Cho 26.54 24.51 24.66 27.39 28.60 30.48 30.33 28.19	EPLL 26.07 25.02 24.23 27.34 28.54 30.39 30.97 28.07	Schmidt 26.32 24.65 24.58 27.36 28.58 30.06 30.17 28.12	Ours 26.92 25.48 24.74 27.92 28.89 31.09 31.39 28.53

4.1. Results

We use a deep DRNN with four hidden layers of size 1024, an input layer of size 961 and an linear output layer of size 121. This model takes a 31×31 blurred patch as input and tries to recover its central 11×11 block as output. We use the natural images in the Berkeley segmentation database [16] and convert them to gray-scale images to generate training samples. Two motion blur kernels provided in [2] are tested and the



Fig. 4. Understanding the representations learned by our model for blur kernel 1. (a) the blurry input patterns learned by the nodes in last hidden layer. (b) the corresponding clean dictionary atoms learned by the output layer. (c) the dictionary in (b) convolved with the blur kernel. Note that they look like the input patterns in (a).



Fig. 3. Improving average PSNR during model training.

Gaussian noise level is $\sigma = 5$. The performances of the results are evaluated using PSNR on eight standard test images. We compare our approach to some state-of-the-art methods, including Krishnan *et al.* [2], Cho *et al.* [3], EPLL [4] and Schmidt *et al.* [5]. For all competing methods, we optimize their hyper-parameters to obtain best possible results. The visual comparisons of the cropped results are shown in Fig. 2. The results of Krishnan *et al.*, Cho *et al.* and Schmidt *et al.* contain some artifacts, while the results of EPLL [4] seem to be overly smoothed. Our models do better in removing the artifacts as well as recovering the details. The quantitative comparisons are listed in Table 1. We can see that our approach outperforms all other methods with significant improvements for both cases.

Training deep neural networks often needs weeks of GPU time [8, 9]. For each training epoch, we use 1.25 million training samples and the training time is about 3 hours of computation time on single Tesla K20c GPU. We carry out 50 epochs to observe the performance evolution. As shown in Fig. 3, the average PSNR improves very slowly after 20 epochs for both cases. Thus about 2.5 days of training time is needed to produce decent results, which makes our model more practical.

4.2. Discussions

As suggested in [17], the multi-layer neural networks are more efficient in signal representation than their shallow counterparts. Thus, using activation maximization method [18], we find the optimal input patterns (of size 31×31) that maximize the activations of the nodes in last hidden layers of our model for blur kernel 1. As shown in Fig. 4(a), these blocks depict some blurry structural information. We also display the dictionary atoms (of size 11×11) learned by these same nodes at the corresponding positions in Fig. 4(b). These blocks represent the weights between the last hidden layer and the output layer, which are used to reconstruct the clean patch. We then convolve these atoms with the blur kernel and get the blurred versions in Fig. 4(c). We can see that they resemble the corresponding input patterns very much.

Based on the observation above, we can make the following analysis. Given a blurred patch as input, our model firstly detects the blurry input patterns from it. Then according to the activations, our model uses the corresponding clean dictionary atoms to reconstruct the deblurred patch as output. This procedure conforms with the discussion in Section 3.2. Thus the deep DRNN model works in image deconvolution by efficiently learning the input patterns and the corresponding reconstruction dictionary.

5. CONCLUSION

In this paper, we have proposed a pure learning approach for non-bind image deconvolution using deep dual-pathway rectifier neural network. We empirically show that our model outperforms the state-of-the-art methods and gain some insight into how our model works by analyzing the learned representations. In the future, we will extend our model to blur kernel estimation and non-uniform deconvolution.

6. REFERENCES

- Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM transactions on graphics* (*TOG*), vol. 26, no. 3, pp. 70, 2007.
- [2] Dilip Krishnan and Rob Fergus, "Fast image deconvolution using hyper-laplacian priors," in Advances in Neural Information Processing Systems, 2009, pp. 1033– 1041.
- [3] Sunghyun Cho, Jue Wang, and Seungyong Lee, "Handling outliers in non-blind image deconvolution," in *IEEE International Conference on Computer Vision*, 2011, pp. 495–502.
- [4] Daniel Zoran and Yair Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE International Conference on Computer Vision*, 2011, pp. 479–486.
- [5] Uwe Schmidt and Stefan Roth, "Shrinkage fields for effective image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2774–2781.
- [6] Junyuan Xie, Linli Xu, and Enhong Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 341–349.
- [7] Viren Jain and Sebastian Seung, "Natural image denoising with convolutional networks," in Advances in Neural Information Processing Systems, 2009, pp. 769–776.
- [8] Harold C Burger, Christian J Schuler, and Stefan Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2392–2399.
- [9] Christian J Schuler, Harold Christopher Burger, Stefan Harmeling, and Bernhard Scholkopf, "A machine learning approach for non-blind image deconvolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1067–1074.
- [10] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia, "Deep convolutional neural network for image deconvolution," in Advances in Neural Information Processing Systems, 2014, pp. 1790–1798.
- [11] Keting Zhang and Liqing Zhang, "Rectifier neural network with a dual-pathway architecture for image denoising," arXiv preprint arXiv:1609.03024, 2016.
- [12] David J Field, "What is the goal of sensory coding?," *Neural computation*, vol. 6, no. 4, pp. 559–601, 1994.

- [13] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce, "Learning a convolutional neural network for nonuniform motion blur removal," in *IEEE Conference* on Computer Vision and Pattern Recognition, 2015, pp. 769–777.
- [14] Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, Quoc V Le, and Andrew Y Ng, "On optimization methods for deep learning," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 265–272.
- [15] Bruno A Olshausen and David J Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [16] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE International Conference on Computer Vision*, 2001, vol. 2, pp. 416–423.
- [17] Yoshua Bengio, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [18] Dumitru Erhan, Aaron Courville, and Yoshua Bengio, "Understanding representations learned in deep architectures," Université de Montréal/DIRO, Technical report, 2010.