

Deep Convolutional Neural Network for Image Deconvolution

Authored by:

Ce Liu
Li Xu
Jimmy SJ Ren
Jiaya Jia

Abstract

Many fundamental image-related problems involve deconvolution operators. Real blur degradation seldom complies with an ideal linear convolution model due to camera noise, saturation, image compression, to name a few. Instead of perfectly modeling outliers, which is rather challenging from a generative model perspective, we develop a deep convolutional neural network to capture the characteristics of degradation. We note directly applying existing deep neural networks does not produce reasonable results. Our solution is to establish the connection between traditional optimization-based schemes and a neural network architecture where a novel, separable structure is introduced as a reliable support for robust deconvolution against artifacts. Our network contains two submodules, both trained in a supervised manner with proper initialization. They yield decent performance on non-blind image deconvolution compared to previous generative-model based methods.

1 Paper Body

Many image and video degradation processes can be modeled as translation-invariant convolution. To restore these visual data, the inverse process, i.e., deconvolution, becomes a vital tool in motion deblurring [1, 2, 3, 4], super-resolution [5, 6], and extended depth of field [7]. In applications involving images captured by cameras, outliers such as saturation, limited image boundary, noise, or compression artifacts are unavoidable. Previous research has shown that improperly handling these problems could raise a broad set of artifacts related to image content, which are very difficult to remove. So there was work dedicated to modeling and addressing each particular type of artifacts in non-blind deconvolution for suppressing ringing artifacts [8], removing noise [9], and dealing with saturated regions [9, 10]. These methods can be further refined

by incorporating patch-level statistics [11] or other schemes [4]. Because each method has its own specialty as well as limitation, there is no solution yet to uniformly address all these issues. One example is shown in Fig. 1 ? a partially saturated blur image with compression errors can already fail many existing approaches. One possibility to remove these artifacts is via employing generative models. However, these models are usually made upon strong assumptions, such as identical and independently distributed noise, which may not hold for real images. This accounts for the fact that even advanced algorithms can be affected when the image blur properties are slightly changed. ? Project webpage: <http://www.lxu.me/projects/dcn/>. The paper is partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 413113).

1

(a)

(b) Krishnan et al .

(c) Ours

Figure 1: A challenging deconvolution example. (a) is the blurry input with partially saturated regions. (b) is the result of [3] using hyper-Laplacian prior. (c) is our result. In this paper, we initiate the procedure for natural image deconvolution not based on their physically or mathematically based characteristics. Instead, we show a new direction to build a data-driven system using image samples that can be easily produced from cameras or collected online. We use the convolutional neural network (CNN) to learn the deconvolution operation without the need to know the cause of visual artifacts. We also do not rely on any pre-process to deblur the image, unlike previous learning based approaches [12, 13]. In fact, it is non-trivial to find a proper network architecture for deconvolution. Previous de-noise neural network [14, 15, 16] cannot be directly adopted since deconvolution may involve many neighboring pixels and result in a very complex energy function with nonlinear degradation. This makes parameter learning quite challenging. In our work, we bridge the gap between an empirically-determined convolutional neural network and existing approaches with generative models in the context of pseudo-inverse of deconvolution. It enables a practical system and, more importantly, provides an empirically effective strategy to initialize the weights in the network, which otherwise cannot be easily obtained in the conventional random-initialization training procedure. Experiments show that our system outperforms previous ones especially when the blurred input images are partially saturated.

2 Related Work Deconvolution was studied in different fields due to its fundamentality in image restoration. Most previous methods tackle the problem from a generative perspective assuming known image noise model and natural image gradients following certain distributions. In the Richardson-Lucy method [17], image noise is assumed to follow a Poisson distribution. Wiener Deconvolution [18] imposes equivalent Gaussian assumption for both noise and image gradients. These early approaches suffer from overly smoothed edges and ringing artifacts. Recent development on deconvolution shows that regularization terms with sparse image priors are important to preserve sharp edges and sup-

press artifacts. The sparse image priors follow heavy-tailed distributions, such as a Gaussian Mixture Model [1, 11] or a hyper-Laplacian [7, 3], which could be efficiently optimized using half-quadratic (HQ) splitting [3]. To capture image statistics with larger spatial support, the energy is further modeled within a Conditional Random Field (CRF) framework [19] and on image patches [11]. While the last step of HQ method is quadratic optimization, Schmidt et al. [4] showed that it is possible to directly train a Gaussian CRF from synthetic blur data. To handle outliers such as saturation, Cho et al. [9] used variational EM to exclude outlier regions from a Gaussian likelihood. Whyte et al. [10] introduced an auxiliary variable in the RichardsonLucy method. An explicit denoise pass is added to deconvolution, where the denoise approach is carefully engineered [20] or trained from noisy data [12]. The generative approaches typically have difficulties to handle complex outliers that are not independent and identically distributed. 2

Another trend for image restoration is to leverage the deep neural network structure and big data to train the restoration function. The degradation is therefore no longer limited to one model regarding image noise. Burger et al. [14] showed that the plain multi-layer perceptrons can produce decent results and handle different types of noise. Xie et al. [15] showed that a stacked denoise autoencoder (SDAE) structure [21] is a good choice for denoise and inpainting. Agostinelli et al. [22] generalized it by combining multiple SDAE for handling different types of noise. In [23] and [16], the convolutional neural network (CNN) architecture [24] was used to handle strong noise such as raindrop and lens dirt. Schuler et al. [13] added MLPs to a direct deconvolution to remove artifacts. Though the network structure works well for denoise, it does not work similarly for deconvolution. How to adapt the architecture is the main problem to address in this paper.

3 Blur Degradation We consider real-world image blur that suffers from several types of degradation including clipped intensity (saturation), camera noise, and compression artifacts. The blur model is given by $y = \text{b}[(x \otimes k + n)]$,

(1)

where x represents the latent sharp image. The notation $x \in \mathbb{R}^n$ is to indicate the fact that x could have values exceeding the dynamic range of camera sensors and thus be clipped. k is the known convolution kernel, or typically referred to as a point spread function (PSF), n models additive camera noise. $\text{b}(\cdot)$ is a clipping function to model saturation, defined as $\text{b}(z) = \min(z, z_{\max})$, where z_{\max} is a range threshold. $\text{b}(\cdot)$ is a nonlinear (e.g., JPEG) compression operator. We note that even with y and kernel k , restoring x is intractable, simply because the information loss caused by clipping. In this regard, our goal is to restore the clipped input x , where $x = \text{b}(x)$. Although solving for x with a complex energy function that involves Eq. (1) is difficult, the generation of blurry image from an input x is quite straightforward by image synthesis according to the convolution model taking all kinds of possible image degradation into generation. This motivates a learning procedure for deconvolution, using training image pairs $\{x_i, y_i\}$, where index $i \in \{1, \dots, N\}$.

4 Analysis The goal is to train a network architecture $f(\cdot)$ that minimizes $\sum_{i=1}^N \|f(y_i) - x_i\|_2^2$, where N is the number of image pairs in the sample set. We have used the recent two deep neural networks to solve this problem, but failed. One is the Stacked Sparse Denoise Autoencoder (SSDAE) [15] and the other is the convolutional neural network (CNN) used in [16]. Both of them are designed for image denoise. For SSDAE, we use patch size 17×17 as suggested in [14]. The CNN implementation is provided by the authors of [16]. We collect two million sharp patches together with their blurred versions in training. One example is shown in Fig. 2 where (a) is a blurred image. Fig. 2(b) and (c) show the results of SSDAE and CNN. The result of SSDAE in (b) is still blurry. The CNN structure works relatively better. But it suffers from remaining blurry edges and strong ghosting artifacts. This is because these network structures are for denoise and do not consider necessary deconvolution properties. More explanations are provided from a generative perspective in what follows.

4.1 Pseudo Inverse Kernels The deconvolution task can be approximated by a convolutional network by nature. We consider the following simple linear blur model $y = x \otimes k$. The spatial convolution can be transformed to a frequency domain multiplication, yielding $F(y) = F(x) \cdot F(k)$.

- (a) input
- (b) SSDAE [15]
- (c) CNN [16]
- (d) Ours

Figure 2: Existing stacked denoise autoencoder and convolutional neural network structures cannot solve the deconvolution problem.

- (a)
- (b)
- (c)
- (d)
- (e)

Figure 3: Pseudo inverse kernel and deconvolution examples. $F(\cdot)$ denotes the discrete Fourier transform (DFT). Operator \cdot is element-wise multiplication. In Fourier domain, x can be obtained as $x = F^{-1}(F(y)/F(k)) = F^{-1}(1/F(k) \cdot y)$, where F^{-1} is the inverse discrete Fourier transform. While the solver for x is written in a form of spatial convolution with a kernel $F^{-1}(1/F(k))$, the kernel is actually a repetitive signal spanning the whole spatial domain without a compact support. When noise arises, regularization terms are commonly involved to avoid division-by-zero in frequency domain, which makes the pseudo inverse falls off quickly in spatial domain [25]. The classical Wiener deconvolution is equivalent to using Tikhonov regularizer [2]. The Wiener deconvolution can be expressed as $x = F^{-1}((F(k)^{-2} + \lambda)^{-1} \cdot F(k)^{-2} \cdot y)$, where λ is the signal-to-noise ratio. k^+ denotes the pseudo inverse kernel. Strong noise leads to a large λ , which corresponds to strongly

regularized inversion. We note that with the introduction of SNR, k^{-1} becomes compact with a finite support. Fig. 3(a) shows a disk blur kernel of radius 7, which is commonly used to model focal blur. The pseudo-inverse kernel k^{-1} with SNR = 10^4 is given in Fig. 3(b). A blurred image with this kernel is shown in Fig. 3(c). Deconvolution results with k^{-1} are in (d). A level of blur is removed from the image. But noise and saturation cause visual artifacts, in compliance with our understanding of Wiener deconvolution. Although the Wiener method is not state-of-the-art, its byproduct that the inverse kernel is with a finite yet large spatial support becomes vastly useful in our neural network system, which manifests that deconvolution can be well approximated by spatial convolution with sufficiently large kernels. This explains unsuccessful application of SSDA and CNN directly to deconvolution in Fig. 2 as follows. SSDA does not capture well the nature of convolution with its fully connected structures. CNN performs better since deconvolution can be approximated by large-kernel convolution as explained above. 4

Previous CNN uses small convolution kernels. It is however not an appropriate configuration in our deconvolution problem. It thus can be summarized that using deep neural networks to perform deconvolution is by no means straightforward. Simply modifying the network by employing large convolution kernels would lead to higher difficulties in training. We present a new structure to update the network in what follows. Our result in Fig. 3 is shown in (e).

5 Network Architecture We transform the simple pseudo inverse kernel for deconvolution into a convolutional network, based on the kernel separability theorem. It makes the network more expressive with the mapping to higher dimensions to accommodate nonlinearity. This system is benefited from large training data. **5.1 Kernel Separability** Kernel separability is achieved via singular value decomposition (SVD) [26]. Given the inverse kernel k^{-1} , decomposition $k^{-1} = U S V^T$ exists. We denote by u_j and v_j the j th columns of U and V , s_j the j th singular value. The original pseudo deconvolution can be expressed as $X s_j = u_j^T (v_j^T y)$, (3) $k^{-1} y = j$

which shows 2D convolution can be deemed as a weighted sum of separable 1D filters. In practice, we can well approximate k^{-1} by a small number of separable filters by dropping out kernels associated with zero or very small s_j . We have experimented with real blur kernels to ignore singular values smaller than 0.01. The resulting average number of separable kernels is about 30 [25]. Using a smaller SNR ratio, the inverse kernel has a smaller spatial support. We also found that an inverse kernel with length 100 is typically enough to generate visually plausible deconvolution results. This is important information in designing the network architecture. **5.2 Image Deconvolution CNN (DCNN)** We describe our image deconvolution convolutional neural network (DCNN) based on the separable kernels. This network is expressed as $h_3 = W_3 h_2$; $h_l = (W_l h_{l-1} + b_{l-1})$, $l \in \{1, 2\}$; $h_0 = y$, where W_l is the weight mapping the $(l-1)$ th layer to the l th one and b_{l-1} is the vector value bias. $\sigma(\cdot)$ is the nonlinear function, which can be sigmoid or hyperbolic tangent. Our network contains two hidden layers similar to the separable kernel inversion setting. The first hidden layer h_1 is generated by applying 38 large-scale one-dimensional

kernels of size 121×1 , according to the analysis in Section 5.1. The values 38 and 121 are empirically determined, which can be altered for different inputs. The second hidden layer h_2 is generated by applying $38 \times 1 \times 121$ convolution kernels to each of the 38 maps in h_1 . To generate results, a $1 \times 1 \times 38$ kernel is applied, analogous to the linear combination using singular value s_j . The architecture has several advantages for deconvolution. First, it assembles separable kernel inversion for deconvolution and therefore is guaranteed to be optimal. Second, the nonlinear terms and high dimensional structure make the network more expressive than traditional pseudo-inverse. It is reasonably robust to outliers.

5.3 Training DCNN The network can be trained either by random-weight initialization or by the initialization from the separable kernel inversion, since they share the exact same structure. We experiment with both strategies on natural images, which are all degraded by additive Gaussian noise (AWG) and JPEG compression. These images are in two categories: one with strong color saturation and one without. Note saturation affects many existing deconvolution algorithms a lot.

Figure 4: PSNRs produced in different stages of our convolutional neural network architecture.

- (a) Separable kernel inversion
- (b) Random initialization
- (c) Separable kernel initialization
- (d) ODCNN output

Figure 5: Results comparisons in different stages of our deconvolution CNN. The PSNRs are shown as the first three bars in Fig. 4. We obtain the following observations. The trained network has an advantage over simply performing separable kernel inversion, no matter with random initialization or initialization from pseudo-inverse. Our interpretation is that the network, with high dimensional mapping and nonlinearity, is more expressive than simple separable kernel inversion. The method with separable kernel inversion initialization yields higher PSNRs than that with random initialization, suggesting that initial values affect this network and thus can be tuned. Visual comparison is provided in Fig. 5(a)-(c), where the results of separable kernel inversion, training with random weights, and of training with separable kernel inversion initialization are shown. The result in (c) obviously contains sharp edges and more details. Note that the final trained DCNN is not equivalent to any existing inverse-kernel function even with various regularization, due to the involved high-dimensional mapping with nonlinearities. The performance of deconvolution CNN decreases for images with color saturation. Visual artifacts could also be yielded due to noise and compression. We in the next section turn to a deeper structure to address these remaining problems, by incorporating a denoise CNN module.

5.4 Outlier-rejection Deconvolution CNN (ODCNN) Our complete network is formed as the concatenation of the deconvolution CNN module with a denoise CNN [16]. The overall structure is shown in Fig. 6. The denoise CNN module has two hidden layers with 512 feature maps. The input image is convolved with 512 kernels of size 16×16 to be fed into the hidden layer. The two network modules are concatenated in our system by combining the last layer of decon-

volution CNN with the input of denoise CNN. This is done by merging the 1×1 kernel with $512 \times 16 \times 16$ kernels to generate 512 kernels of size $16 \times 16 \times 36$. Note that there is no nonlinearity when combining the two modules. While the number of weights grows due to the merge, it allows for a flexible procedure and achieves decent performance, by further incorporating fine tuning. 6

64x184x38
49x49x512
64x64x38
49x49x512
184x184
kernel size 1×121
56x56
kernel size 121×1
kernel size $16 \times 16 \times 38$
kernel size $1 \times 1 \times 512$
kernel size $8 \times 8 \times 512$
Outlier Rejection Sub-Network
Deconvolution Sub-Network
Restoration

Figure 6: Our complete network architecture for deep deconvolution.

5.5 Training ODCNN We blur natural images for training ? thus it is easy to obtain a large number of data. Specifically, we use 2,500 natural images downloaded from Flickr. Two million patches are randomly sampled from them. Concatenating the two network modules can describe the deconvolution process and enhance the ability to suppress unwanted structures. We train the sub-networks separately. The deconvolution CNN is trained using the initialization from separable inversion as described before. The output of deconvolution CNN is then taken as the input of the denoise CNN. Fine tuning is performed by feeding one hundred thousand 184×184 patches into the whole network. The training samples contain all patches possibly with noise, saturation, and compression artifacts. The statistics of adding denoise CNN are also plotted in Fig. 4. The outlier-rejection CNN after fine tuning improves the overall performance up to 2dB, especially for those saturated regions.

6 More Discussions Our approach differs from previous ones in several ways. First, we identify the necessity of using a relatively large kernel support for convolutional neural network to deal with deconvolution. To avoid rapid weight-size expansion, we advocate the use of 1D kernels. Second, we propose a supervised pre-training on the sub-network that corresponds to reinterpretation of Wiener deconvolution. Third, we apply traditional deconvolution to network initialization, where generative solvers can guide neural network learning and significantly improve performance. Fig. 6 shows that a new convolutional neural network architecture is capable of dealing with deconvolution. Without a good understanding of the functionality of each sub-net and performing supervised pre-training, however, it is difficult to make the network work very well. Training the whole network with random initialization is less preferred because the

training algorithm stops halfway without further energy reduction. The corresponding results are similarly blurry as the input images. To understand it, we visualize intermediate results from the deconvolutional CNN sub-network, which generates 38 intermediate maps. The results are shown in Fig. 7, where (a) is the selected three results obtained by random-initialization training and (b) is the results at the corresponding nodes from our better-initialized process. The maps in (a) look like the high-frequency part of the blurry input, indicating random initialization is likely to generate high-pass filters. Without proper starting values, its chance is very small to reach the component maps shown in (b) where sharper edges present, fully usable for further denoise and artifact removal. Zeiler et al. [27] showed that sparsely regularized deconvolution can be used to extract useful middle-level representation in their deconvolution network. Our deconvolution CNN can be used to approximate this structure, unifying the process in a deeper convolutional neural network. 7

(a)
(b)

Figure 7: Comparisons of intermediate results from deconvolution CNN. (a) Maps from random initialization. (b) More informative maps with our initialization scheme. kernel type Krishnan [3] Levin [7] disk sat. 24.05dB 24.44dB disk 25.94dB 24.54dB motion sat. 24.07dB 23.58dB motion 25.07dB 24.47 dB

Cho [9] Whyte [10] Schuler [13] Schmidt [4] 25.35dB 24.47dB 23.14dB 24.01dB 23.97dB 22.84dB 24.67dB 24.71dB 25.65 dB 25.54dB 24.92dB 25.33dB 24.29dB 23.65dB 25.27dB 25.49dB

Ours 26.23dB 26.01dB 27.76dB 27.92dB

Table 1: Quantitative comparison on the evaluation image set.

(a) Input
(b) Levin et al. [7]
(c) Krishnan et al. [3]
(d) EPLL [11]
(e) Cho et al. [9]
(f) Whyte et al. [10]
(g) Schuler et al. [13]
(h) Ours

Figure 8: Visual comparison of deconvolution results.

7 Experiments and Conclusion We have presented several deconvolution results. Here we show quantitative evaluation of our method against state-of-the-art approaches, including sparse prior deconvolution [7], hyperLaplacian prior method [3], variational EM for outliers [9], saturation-aware approach [10], learning based approach [13] and the discriminative approach [4]. We compare performance using both disk and motion kernels. The average PSNRs are listed in Table 1. Fig. 8 shows a visual comparison. Our method achieves decent results quantitatively and visually. The implementation, as well as the dataset, is available at the project webpage. To conclude this paper, we have proposed a new deep convolutional network structure for the challenging image deconvolution task. Our main contribution is to let traditional deconvolution schemes guide neural networks and approximate deconvolution by a series of

convolution steps. Our system novelly uses two modules corresponding to deconvolution and artifact removal. While the network is difficult to train as a whole, we adopt two supervised pre-training steps to initialize sub-networks. High-quality deconvolution results bear out the effectiveness of this approach.

2 References

- [1] Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM Trans. Graph.* 25(3) (2006) 8
- [2] Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: *CVPR*. (2009) [3] Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: *NIPS*. (2009) [4] Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., Roth, S.: Discriminative non-blind deblurring. In: *CVPR*. (2013) [5] Agrawal, A.K., Raskar, R.: Resolving objects at higher resolution from a single motion-blurred image. In: *CVPR*. (2007) [6] Michaeli, T., Irani, M.: Nonparametric blind super-resolution. In: *ICCV*. (2013) [7] Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.* 26(3) (2007) [8] Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graph.* 27(3) (2008) [9] Cho, S., Wang, J., Lee, S.: Handling outliers in non-blind image deconvolution. In: *ICCV*. (2011) [10] Whyte, O., Sivic, J., Zisserman, A.: Deblurring shaken and partially saturated images. In: *ICCV Workshops*. (2011) [11] Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: *ICCV*. (2011) [12] Kenig, T., Kam, Z., Feuer, A.: Blind image deconvolution using machine learning for three-dimensional microscopy. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(12) (2010) [13] Schuler, C.J., Burger, H.C., Harmeling, S., Schölkopf, B.: A machine learning approach for non-blind image deconvolution. In: *CVPR*. (2013) [14] Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: *CVPR*. (2012) [15] Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: *NIPS*. (2012) [16] Eigen, D., Krishnan, D., Fergus, R.: Restoring an image taken through a window covered with dirt or rain. In: *ICCV*. (2013) [17] Richardson, W.: Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America* 62(1) (1972) [18] Wiener, N.: Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications. *Journal of the American Statistical Association* 47(258) (1949) [19] Roth, S., Black, M.J.: Fields of experts. *International Journal of Computer Vision* 82(2) (2009) [20] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: Image restoration by sparse 3d transform-domain collaborative filtering. In: *Image Processing: Algorithms and Systems*. (2008) [21] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of*

Machine Learning Research 11 (2010) [22] Agostinelli, F., Anderson, M.R., Lee, H.: Adaptive multi-column deep neural networks with application to robust image denoising. In: NIPS. (2013) [23] Jain, V., Seung, H.S.: Natural image denoising with convolutional networks. In: NIPS. (2008) [24] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11) (1998) [25] Xu, L., Tao, X., Jia, J.: Inverse kernels for fast spatial deconvolution. In: ECCV. (2014) [26] Perona, P.: Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 17(5) (1995) [27] Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: CVPR. (2010)