

Iterative Residual Image Deconvolution

Li Si-Yao¹, Dongwei Ren², Furong Zhao³ Zijian Hu¹, Junfeng Li¹, Qian Yin^{1*}

¹ Beijing Normal University ² Tianjin University ³ Sensetime Research
 lisiyao@mail.bnu.edu.cn, {rendongweihit, hanshzj1998}@gmail.com,
 zhaofurong@sensetime.com, {lijunfeng, yinqian}@bnu.edu.cn

Abstract

Image deblurring, a.k.a. image deconvolution, recovers a clear image from pixel superposition caused by blur degradation. Few deep convolutional neural networks (CNN) succeed in addressing this task. In this paper, we first demonstrate that the minimum-mean-square-error (MMSE) solution to image deblurring can be interestingly unfolded into a series of residual components. Based on this analysis, we propose a novel iterative residual deconvolution (IRD) algorithm. Further, IRD motivates us to take one step forward to design an explicable and effective CNN architecture for image deconvolution. Specifically, a sequence of residual CNN units are deployed, whose intermediate outputs are then concatenated and integrated, resulting in concatenated residual convolutional network (CRCNet). The experimental results demonstrate that proposed CRCNet not only achieves better quantitative metrics but also recovers more visually plausible texture details compared with state-of-the-art methods.

Introduction

Image deblurring that aims at recovering a clear image from its blurry observation receives considerable research attention in decades. The blurry image b is usually modeled as a convolution of clear image x and blur kernel k , i.e.,

$$b = k * x + \eta, \quad (1)$$

where $*$ denotes 2D convolution, and η is additive noise. Thus, image deblurring is also well known as image deconvolution (Andrews and Hunt 1977; Kundur and Hatzinakos 1996). When blur kernel k is given, clear images can be recovered by deconvolution under the maximum a posterior (MAP) framework (Andrews and Hunt 1977; Fergus et al. 2006; Levin et al. 2007; Krishnan and Fergus 2009):

$$\hat{x} = \arg \min_x (\|k * x - b\|^2 + \lambda \mathcal{R}(x)), \quad (2)$$

where $\mathcal{R}(x)$ is regularization term associated with image prior, and λ is a positive trade-off parameter.

In conventional deconvolution methods, considerable research attention is paid on the study of regularization term for better describing natural image priors, including Total Variation (TV) (Wang et al. 2008), hyper-Laplacian (Krishnan and Fergus 2009), dictionary sparsity (Zhang et al. 2010; Hu, Huang, and Yang 2010), non-local similarity (Dong,

Shi, and Li 2013), patch-based low rank prior (Ren et al. 2016) and deep discriminative prior (Li et al. 2018). Note that alternative direction method of multipliers (ADMM) is often employed to efficiently solve these models. Besides, driven by the success of discriminative learning, the image priors can be learned from abundant training samples. With the half-quadratic splitting strategy, regression tree field (Jancsary et al. 2012; Schmidt et al. 2013) and shrinkage field (Schmidt and Roth 2014) are proposed to model regularization term, and are effectively trained stage-by-stage. These learning-based methods have validated the superiority of discriminative learning over manually selected regularization term (Schmidt et al. 2013; Schmidt and Roth 2014).

Most recently, deep convolutional neural network (CNN), as a general approximator, has been successfully applied in low level vision tasks, e.g., image denoising (Zhang et al. 2017), inpainting (Yang et al. 2017), superresolution (Dong et al. 2014). As for image deblurring, there are also several attempts, in which CNN is used to directly map blurry images to clear ones. In (Nah, Hyun Kim, and Mu Lee 2017), a deep multi-scale CNN is designed in image deblurring without explicit blur kernel estimation; as an upgrade, an recurrent unit is embedded into CNN such that multi scales share same CNN weight (Tao et al. 2018). In (Kupyn et al. 2018), a generative adversarial network (GAN) tries to train a ResNet (He et al. 2016) supervised by the adversarial discriminator. However, these trained CNN-based models can only handle mildly blurry images, and usually fail in real cases, since practical blur trajectories are complex.

When the blur kernel is known, CNN-based deconvolution has also been studied. On one hand, Xu *et al* (2014). have validated that plain CNN cannot succeed in deconvolution. To make CNN work well for deconvolution, a specific blur kernel would be decomposed into inverse kernels (Xu, Tao, and Jia 2014), which are then used to initialize CNN, inevitably limiting its practical applications. On the other hand, CNN is incorporated into conventional deconvolution algorithms under plug-and-play strategy. In (Kruse, Rother, and Schmidt 2017), CNN is employed to solve denoising subproblem in ADMM modulars. In (Zhang et al. 2017), CNN-based Gaussian denoisers are trained off-line, and are iteratively plugged under half-quadratic strategy. Although these methods empirically achieve satisfactory results, they

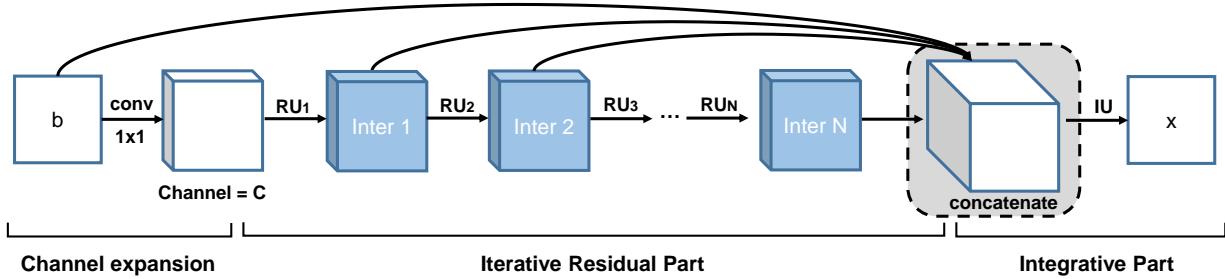


Figure 1: The architecture of CRCNet. In this demonstration, RU_i is the i -th Residual Unit and IU represents the Integrative Unit.

are not trained end-to-end, and some parameters need to be tuned for balancing CNN strength. As a summary, the effective CNN architecture for deconvolution still remains unsolved.

In this paper, we propose a novel concatenated residual convolutional network (CRCNet) for deconvolution. We first derive a closed-form deconvolution solution driven by the minimum mean square error (MMSE)-based discriminative learning. Then, using a power series expansion, we unfold MMSE solution into a sum of residual convolutions, which we name iterative residual deconvolution (IRD) algorithm. IRD is a very simple yet effective deconvolution scheme. As shown in Figure 2, the blur can be effectively removed with the increasing of iterations. Although IRD would magnify the noise in degraded image, the blur could still be significantly removed. Motivated by this observation, we design an effective CNN architecture for deconvolution, as shown in Figure 1. We adopt residual CNN unit to substitute the residual component in IRD. These residual CNN units are iteratively connected, and all intermediate outputs are concatenated and finally integrated, resulting in CRCNet. Interestingly, the developed non-linear CRC model behaves efficient and robust. On test datasets, CRCNet can achieve higher quantitative metrics and recover more visually plausible texture details from compared with state-of-the-art algorithms. We claim that effective CNN architecture plays the critical role in deconvolution, and CRCNet is one of the successful attempts. Our contributions are three-fold:

- We derive a closed-form deconvolution solution driven by MMSE-based discriminative learning, and further unfold it into a series, as a simple yet effective IRD algorithm.
- Motivated by IRD algorithm, we propose a novel CRCNet for deconvolution. The CRCNet can be trained end-to-end, but is not a plain CNN architecture and is well analyzed.
- We discuss the contributions of CRCNet, and show the critical role of network architecture for deconvolution. Experimental results demonstrate the superiority of CRCNet over state-of-the-art algorithms.

The remainder of this paper is organized as follows: Section 2 derives MMSE-based deconvolution, and then presents IRD algorithm. Section 3 designs CRCNet based

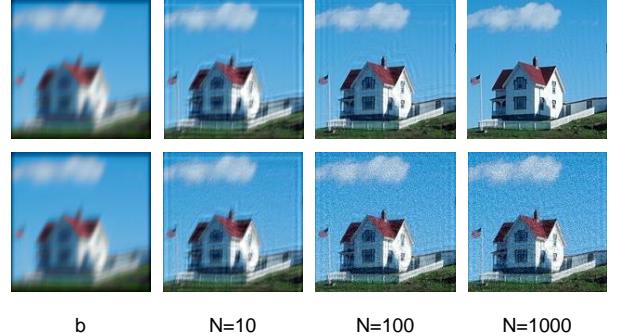


Figure 2: Deblurring results using $H^T \sum_{n=0}^N (\sigma' I - HH^T)^n$. The first row: deblurring image without noise; the second row: deblurring noisy image with $\sigma = 0.01$. Here pixels of natural images are assumed to be independent and $C_x = I$. For visualization, the deblurring is operated on the first channel of YCbCr.

on IRD, along with its training strategy. Section 4 demonstrates experimental results and Section 5 ends this paper with conclusions.

Iterative Residual Deconvolution

In this section, we first derive a deconvolution solution driven by minimum mean square error (MMSE) (Andrews and Hunt 1977), which is then unfolded via series expansion, resulting in iterative residual deconvolution (IRD) algorithm. Finally we give an insightful analysis to IRD, and provide a potential CNN architecture for deconvolution.

MMSE Deconvolution

The convolution operation in Eqn. (1) can be equivalently reformatted as a linear transform

$$k * x = Hx, \quad (3)$$

where H is a Blocked Toeplitz Matrix (Andrews and Hunt 1977) and \underline{A} represents the column-wise expansion vector of matrix A . Then we aim to seek a linear transform L to recover clear image

$$\hat{x} = Lb. \quad (4)$$

Let us assume a set of training image pairs $\{x, b\}$. By minimizing MSE loss, we have

$$\begin{aligned} L &= \arg \min_L \mathbb{E}_x [\text{Tr}(\underline{x} - \hat{x})(\underline{x} - \hat{x})^T] \\ &= C_x H^T (H C_x H^T + C_\eta)^{-1}, \end{aligned} \quad (5)$$

where $C_x = x x^T$ and $C_\eta = \eta \eta^T$ are Gramm matrices of clear images and noises, respectively. $C_x[i, j]$ represents the correlation between the i -th pixel and the j -th pixel of a sharp image and C_η is similar.

On one hand, correlations among pixels of natural images are limited (Hu, Xue, and Zheng 2012), thus eigenvalues of C_x can be deemed to be positive. Then, for any possible C_x , we can always find an $\alpha > 0$ such that $\lambda_{\max}(\alpha C_x) \leq 1$ where λ_{\max} represents the greatest eigenvalue. Hence, we have

$$\begin{aligned} L &= C_x H^T ((1/\alpha)(H \alpha C_x H^T + \alpha C_\eta))^{-1} \\ &= C'_x H^T (H C'_x H^T + C'_\eta)^{-1}, \end{aligned} \quad (6)$$

where $C'_x = \alpha C_x$ and $C'_\eta = \alpha C_\eta$.

On the other hand, η is deemed to be zero-meaned gaussian noise with strength σ/α (assumed small in this work), thus C'_η approximates to σI . Hence, L can be approximated as

$$L \approx C'_x H^T (H C'_x H^T + \sigma I)^{-1}. \quad (7)$$

To now, we have obtained a closed-form solution for deconvolution.

IRD Algorithm

For matrix A with $A^n \rightarrow 0$ as $n \rightarrow \infty$, the following series expansion holds:

$$(I - A)^{-1} = \sum_{n=0}^{\infty} A^n. \quad (8)$$

As for the case in deconvolution, blur kernel k is under two constraints (Kundur and Hatzinakos 1996; Levin et al. 2009; Perrone and Favaro 2014):

$$k_{ij} \geq 0 \quad (9)$$

and

$$\sum_{i,j} k_{ij} = 1. \quad (10)$$

Under such constraints, the norm of degradation matrix H is limited under 1. We also found empirically that eigenvalues of $H H^T$ are generally positive.

Thus, the linear deconvolution solution L in Eqn. (7) can be unfolded as follows:

$$L \approx [C'_x H^T] \left[\sum_{n=0}^N (\sigma' I - H C'_x H^T)^n \right] \quad (11)$$

where $\sigma' = 1 - \sigma$. Eqn. (11) can be implemented as an iterative algorithm. Matrix multiplications by H and H^T are equal to convolutions with k and the flipped kernel k_- .¹ The

¹The flip operation corresponds to $\text{rot90}(\cdot, 2)$ in MatLab.

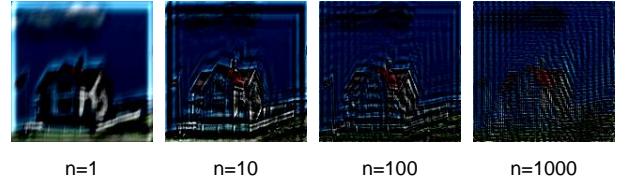


Figure 3: Intermediate components $H^T(\sigma' I - H H^T)^n b$ corresponding to the first row in Figure 2 with increasing n . The operation is taken on the first channel of YCbCr. The original pixel values are too small when n is large, thus images are scaled by 1.5, 10^2 , 10^3 , 2×10^4 respectively to keep the maximum equal to 1 for visualization.

Algorithm 1 Iterative Residual Deconvolution

Input: blurry image b , degradation kernel k , image priori patch f_x , noise strength σ
Output: restored \hat{x}

```

1:  $\sigma' \leftarrow 1 - \sigma$ ,  $s \leftarrow b$ ,  $r \leftarrow b$ 
2: for  $i \leftarrow 0$  to  $N - 1$  do
3:    $r \leftarrow \sigma' r - k * f_x * k_- * r$ 
4:    $s \leftarrow s + r$ 
5: end for
6:  $\hat{x} \leftarrow f_x * k_- * s$ 

```

correlation matrix C'_x actually plays as the prior of clear images and is assured to be Toeplitz (Andrews and Hunt 1977). Hence, linear transform C'_x is equivalent to a convolution with a limited patch f_x for pixels of clear images are correlated only in the vicinity. By assuming pixels in a clear image are independent (Hu, Xue, and Zheng 2012; Ren et al. 2018), C'_x can be simplified as identity matrix I , i.e., $f_x = \delta$. The detailed process is summarized as IRD Algorithm 1.

The IRD algorithm is very simple yet effective for deconvolution. Figure 2 shows that the clear image can be satisfactorily restored from a noise-free blurry one after 1000 iterations. Although the noise is significantly magnified for a noisy blurry image, the blur can also be effectively removed.

To explore the significance of unfolded components, we extracted $C'_x H^T (\sigma' I - H C'_x H^T)^n$ as shown in Figure 3. The energy of iterative residues attenuates but the component represents more detailed signals in higher frequency with increasing n . Each iteration extracts residual information from the result of the previous iteration, and those components are finally summed to a clear image.

Comparison to Other Unfolded Methods

The proposed IRD is different from the existing unfolded algorithms. Previous unfolded methods focus on optimization to Eqn. (2). Specifically, ADMM introduces auxiliary variabel z and augmented Lagrange multiplier y into the original object function and optimizes each variable alternately. As another popular iterative deblur scheme, the accelerated proximal gradient (APG, also named as Fast Iterative Shrinkage/Thresholding Algorithm, FISTA) updates a dual variable to the proximal gradient mapping of (2), which

Algorithm 2 simplified L1-regularized ADMM

Input: blurry image b , degradation matrix H , trade-off ρ
Output: restored x

- 1: **while** not converge **do**
- 2: $x \leftarrow (H^T H + \rho I)^{-1} (H^T b + \rho z - y)$
- 3: $z \leftarrow S_{\lambda/\rho}(x + y/\rho)$
- 4: $y \leftarrow y + \rho(x - z)$
- 5: **end while**

Algorithm 3 simplified L1-regularized APG

Input: blurry image b , degradation matrix H , trade-off ρ
Output: restored x

- 1: **while** not converge **do**
- 2: $x^{(i)} \leftarrow S_{\lambda t}(y - 2\lambda t H^T (Hy - b))$
- 3: $y \leftarrow x^{(i)} + \frac{i-1}{i+2}(x^{(i)} - x^{(i-1)})$
- 4: $i \leftarrow i + 1$
- 5: **end while**

significantly accelerates the convergence. A simplified L1-regularized version of ADMM and APG is shown in Algorithm 2 and 3.²

In contrast, IRD reformulates the inverse process into residual convolutions and represents MMSE deconvolution as a sum of image components with gradually increasing frequency but lower energy. More interestingly, IRD provides a potential network structure for deconvolution. The iterative residual deconvolution pipeline reminds us the residual learning structure proposed by He *et al.* (2016). All convolutional parts in IRD can be learned as weights of a CNN. Such an analogy inspired us to propose the following network structure.

Concatenated Residual Convolutional Network

By imitating IRD algorithm, we designed a network as shown in Figure 1, which includes two main parts: the Iterative Residual Part and the Integrative Part, corresponding to $[(\sigma' I - HC'_x H^T)^n]$ and $[C'_x H^T \sum_{n=0}^N]$, respectively. For the first part, $\sigma' I - HC'_x H^T$ corresponds to the *conv-deconv-minus* structure of a Residual Unit. Considering that linear operator C_x is symmetric $C_x = C_x^{\frac{1}{2}} C_x^{\frac{1}{2}T}$, $HC_x H^T$ can be separated into $(HC_x^{\frac{1}{2}})(C_x^{\frac{1}{2}T} H^T)$. Note that operator H and C_x are equivalent to convolutions (see section 2), so their transposes correspond to transpose convolutions (also called *deconvolutions* in CNN). For the second, operator $[C'_x H^T \sum_{n=0}^N]$ is implemented as *conv* layers on the concatenation of all residues with gradually decreased channels. Because a CNN manipulates convolutions channel-wisely and sum the convolutions of all channels, this structure can sum all residues while adopting convolutions.

We take parametric rectified linear units (PReLU) (He *et al.* 2015) between *conv* or *deconv* layers. The slope of

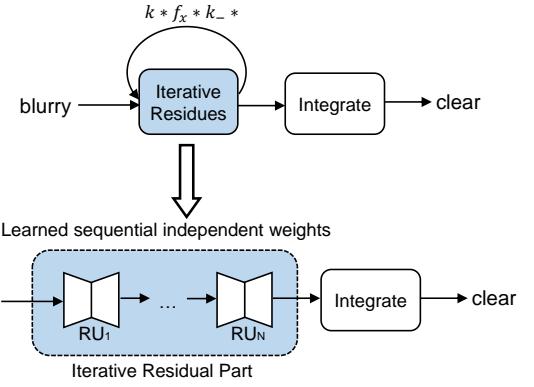


Figure 4: Illustration of IRD to CRCNet: iterative shared weights to sequential isolated layers.

PReLU on negative part is learned during backpropagation, which can be deemed as a non-linear expansion to IRD.

Channel Expansion. As the first step of CRCNet deblurring, the channel of input blurry image is mapped from 1 to C through a 1×1 *conv* layer. The destination of channel expansion is to enhance the capability of *conv/deconv* weights and hence to improve network's flexibility.

Residual Unit. A Residual Unit (RU) calculates the difference between the input and the processed image to extract valid information. Formally,

$$RU(x) = x - deconv(PReLU(conv(x))). \quad (12)$$

Compared to the Eqn. (11), in each RU, convolutional and deconvolutional (transpose convolutional) layers resemble $HC'_x H^T$. However, the auto-encoder-like network can realize more complicated transforms by taking advantage of non-linearity layers. Further, the weights of convolutional layers are learned from not only the blur but also clear images. Hence, an RU can extract information of images more efficiently.

Iterative Residual Part. The intermediate output of a Residual Unit is fed to the next iteratively. As shown in Figure 1, RU_i represents the i -th RU, and $inter_i$ is the output of RU_i . Formally,

$$inter_i = RU_i(inter_{i-1}), \quad (13)$$

where $inter_0$ is extended blurry input b .

Integration. The last part of our network is to integrate all extracted information from the blurry image. The input b and all intermediate residues are concatenated and fed into an Integrative Unit (IU). IU takes three 7×7 *conv* layers to play the role of $[C'_x H^T]$ and the channel dimension decreases gradually to 1 through convolutions as a weighted sum of unfolded components.

² S_α is the soft shrinkage function at α

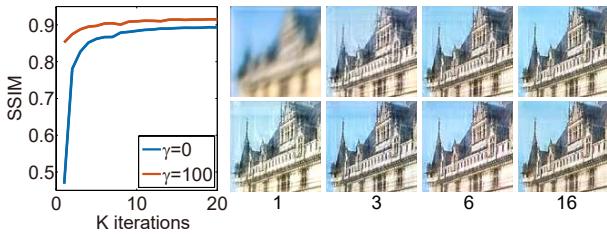


Figure 5: Quantitative and visual comparisons on deblurring a test image with and without taking edge loss. Taking edge loss not only accelerates the training process but also enhances the quality of restored images.

Loss Function. An ideal deblurring model is expected to restore sufficient content of clear image x and make the restored \hat{x} looks sharp. Thus, the loss function of our network is designed to consist of a *content loss* and an *edge loss*:

$$\mathcal{L} = \alpha \mathcal{L}_c + \gamma \mathcal{L}_e, \quad (14)$$

where \mathcal{L}_c is the smooth L_1 loss (Girshick 2015):

$$\mathcal{L}_c(\hat{x}, x) = \text{smooth}_{L_1}(\hat{x} - x), \quad (15)$$

which is more robust on outliers than MSE loss, and

$$\mathcal{L}_e(\hat{x}, x) = \|\partial_h \hat{x} - \partial_h x\|^2 + \|\partial_v \hat{x} - \partial_v x\|^2, \quad (16)$$

in which ∂_h and ∂_v represent horizontal and vertical differential operator.

The *edge loss* constrains edges of \hat{x} to be close to those of x . Our experiment showed that adding \mathcal{L}_e could speed up the convergence of the network efficiently and make restored edges sharp (See Figure 5).

Experimental Results

Training CRCNet

Training Dataset Preparation *Clear Image Set.* Clear images are essential to train network weights. The dataset is expected to only consist of uniformly sharp and noise-free images with ample textures. We manually selected and clipped 860 256×256 RGB images from BSD500 (Martin et al. 2001) and COCO (Lin et al. 2014) dataset, during which we omitted all pictures with Bokeh Effect or motion blur.

Degradation Kernels. We randomly generated 10 21×21 degradation kernels by using code from (Chakrabarti 2016) for training and testing. The generated kernels are shown in Figure 6.

Training details We cropped training images into 35×35 patches and used Adam (Kingma and Ba 2014) optimizer with a mini-batch of size 32 for training. The initial learning rate is 10^{-4} and decay 0.8 per 1000 iterations. The network was only trained 20K iterations for each blur kernel to keep this process portable. In our experiment $\alpha = 5000$ and $\gamma = 100$. Ten clear images with ample details were selected for tests and the rest 850 were used for training.

In our experiments, expanded channel $C = 10$. Kernel sizes of *conv* and *deconv* of each RU are 7×7 . We take

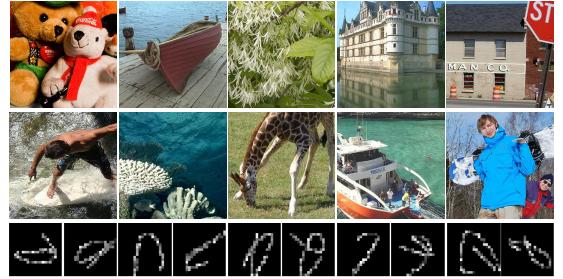


Figure 6: Proposed clear image set and degradation kernels.

$N = 10$ in iterative residual part and the dimension of the final concatenation before integration is 101.

The CRCNet was implemented in Python with PyTorch and tested on a computer with GeForce GTX 1080 Ti GPU and Intel Core i7-6700K CPU.

Comparison to states of the art

Test on synthetic blurry images. Quantitative evaluations of average PSNR and SSIM of 10 test images with 10 kernels are shown in Table 1. Several test results are shown in Figure 7. Compared with state-of-art approaches including traditional MAP method using Hyper-Laplacian priors (HL) (Krishnan and Fergus 2009), learning-based method CSF (Schmidt and Roth 2014) and CNN-based methods IRCNN (Zhang et al. 2017) and FDN (Kruse, Rother, and Schmidt 2017), CRCNet recovers more details in restored images, e.g., fur of Teddy bears and bright spray. Thus deblurred results of our method look more natural and vivid.

Specifically among contrasts, Schmidt and Roth (2014) substitute the shrinkage mapping of half-quadratic (similar to ADMM but without the Lagrange multiplier) into a learned function constituted by multiple Gaussians, which could be deemed as a learning expansion to gradient-based unfolded method. CRCNet, as a derivative from IRD, achieves better performance. The current literature lacks an learning-expanded deblurring method of APG, thus we don't list relative methods into comparison.

We also tested our network on benchmark Levin's set (Levin et al. 2009), which concludes 4 test images \times 8 kernels. The quantitative results are shown in Table 1 and visual comparisons are in Figure 8. Our approach achieves higher performance both visually and quantitatively.

Test on real blurry images. We test proposed CRCNet and state-of-the-art methods on real-world blurry images. These blurry images are produced by superposing 16 adjoining frames in motion captured using GoPro Hero 6. Blur kernels are estimated by (Zuo et al. 2015). Figure 10 shows that previous methods result in strong ringing effect and hence lower the image quality. In contrast, CRCNet remains plausible visual details while avoiding artifacts. We also take a quantitative perceptual scores proposed in (Ma et al. 2017) on all methods; CRCNet obtains the highest (see Table 3).

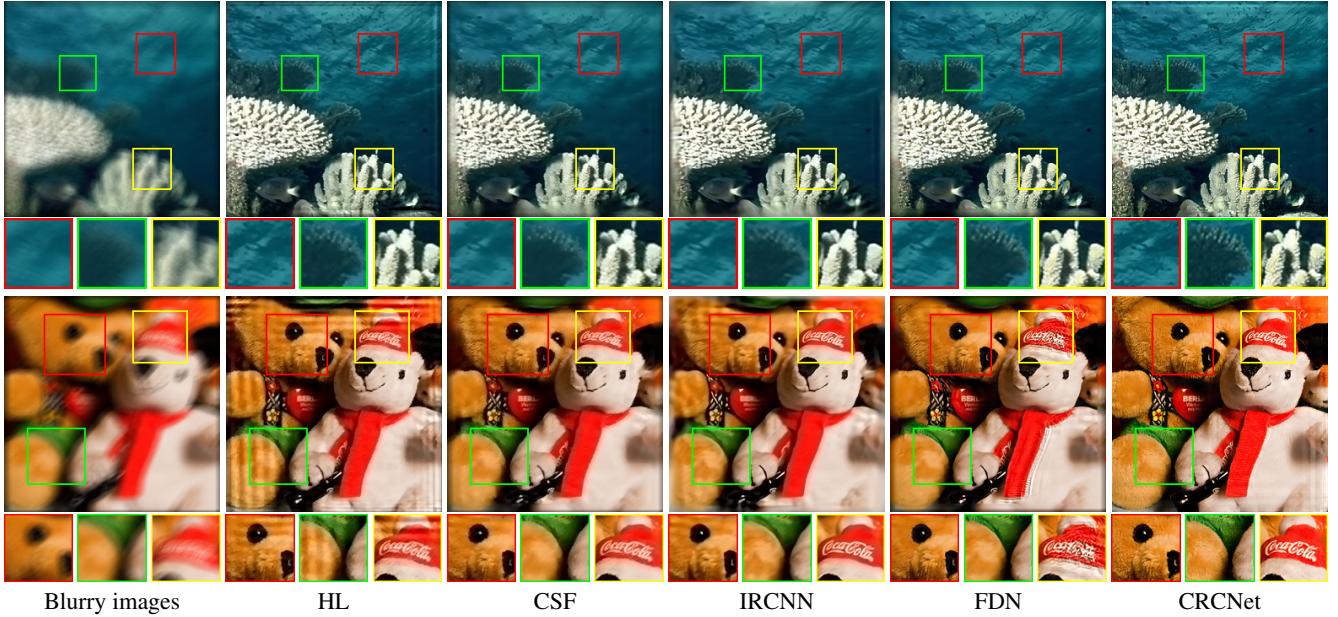


Figure 7: Visual comparison of deblurring results with state-of-art approaches.

Test set	Levin	Proposed
HL	30.20 / 0.90	23.56 / 0.77
CSF	33.53 / 0.93	27.24 / 0.85
IRCNN	34.68 / 0.94	27.28 / 0.84
FDN	35.08 / 0.96	29.37 / 0.89
CRCNet	35.39 / 0.96	29.83 / 0.92

Table 1: Quantitative evaluations of different deblurring methods using ground truth kernel on corresponding test set. Each number pair notes the mean PSNR/SSIM scores.

Method	DCNN	CRCNet
PSNR / SSIM	26.77 / 0.85	27.05 / 0.86
param. amount	15M	0.4M

Table 2: Quantitative evaluations of DCNN and CRCNet.

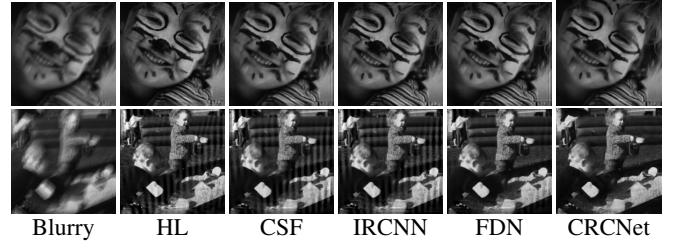


Figure 8: Visual comparison of deblurring results on Levin's set with state-of-art approaches.



Figure 9: Deblurring disk7 examples using DCNN and CRCNet.

Comparison with DCNN. In the last part of experiments, we compare our method with previous non-blind deconvolution network DCNN on accompanying dataset in (Xu et al. 2014) (see Table 2 and Figure 9). This comparison is listed separately for only test code and trained weights on disk7 of DCNN are published. In this experiment, kernel is limited as uniform disk of 7-pixel radius and blurry images are extra degraded by saturation and lossy compression. We also list weight amounts of both networks. CRCNet obtains higher performance while taking much less network parameters. Further, DCNN requires specific initializations while CRCNet can be trained directly in end-to-end way.

The implementation of this work and the clear image set are published at <https://github.com/lisiyaoATbnu/crcnet>.

Analysis to CRCNet

A question beyond the superior performance is whether the effectiveness of CRCNet depends on our proposed concatenated residual (CR) architecture or just a trivial ‘universal approximator’ relying on neural networks. To verify the contribution of CR structure, we give a discussion on relationship between IRD and CRCNet.

CRCNet plays a sequential nonlinear expansion of the iterative structure of IRD. Specifically, CRCNet realizes iterative residues by several learned isolated *conv/deconv* layers rather than thousands of iterations using fixed shared weights k and f_x in IRD (see Figure 4). This iterative-to-

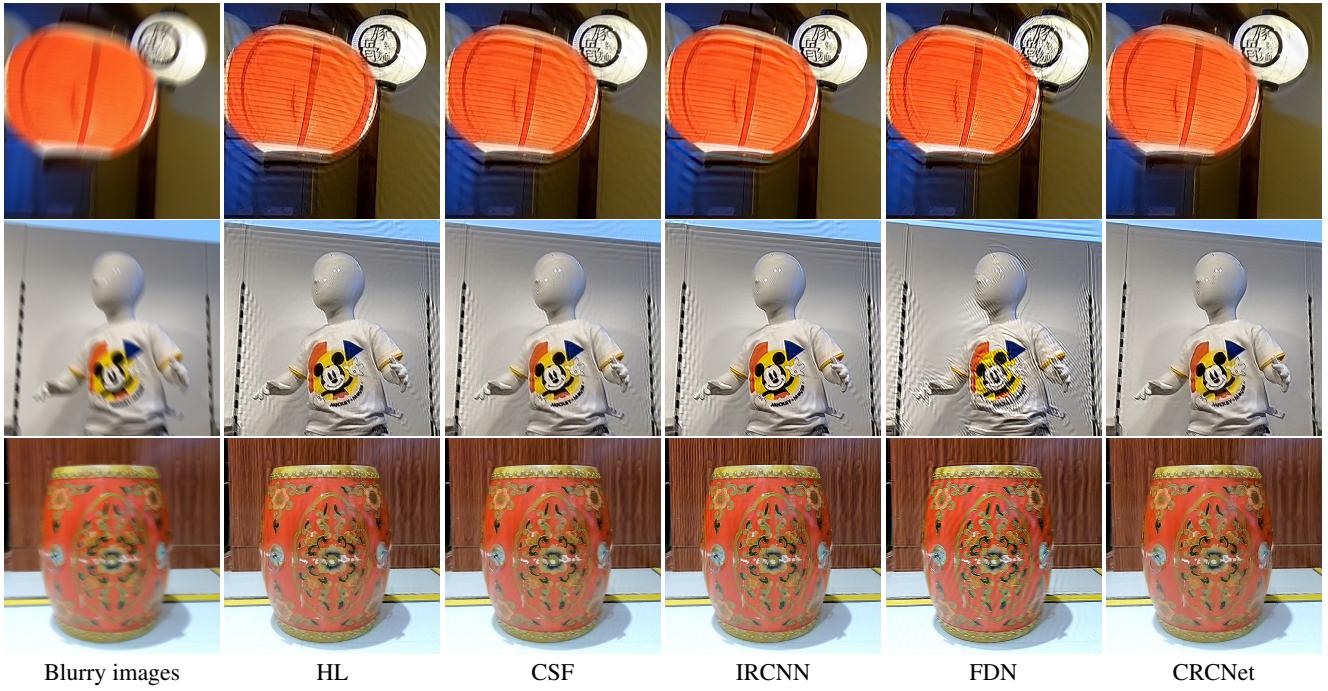


Figure 10: Visual comparison of deblurring results with state-of-art approaches on three real blurry images. From top to bottom are named as lantern, model and chair.

Test image	lantern	model	chair
Blurry	3.3	3.2	3.5
HL	8.6	7.4	8.3
CSF	8.0	6.4	7.7
IRCNN	8.3	6.8	5.1
FDN	8.4	6.3	6.3
CRCNet	8.7	8.5	8.7

Table 3: Quantitative evaluations of different deblurring methods on real blurry images. Each number notes the corresponding perceptual score proposed in (Ma et al. 2017).

sequential expansion enhances the flexibility and capacity of original method. In IRD algorithm, a large number of iterations are required for satisfactory deblurring quality; but in CRCNet, due to the powerful modeling capability of CNN, a very small number of layers can provide good restoration quality.

The iterative residual structure drives CRCNet to process images like IRD. To illustrate this point, we visualized intermediate outputs $inter_i$. Figure 11 shows that deep outputs of CRCNet contain high-frequency oscillations along edges in the image. That fact actually resembles IRD algorithm extracting high-frequency details after large amounts of iterations, as shown in Figure 3.

We in this paper claim that deep CNN-based model for deconvolution should be equipped with specific architecture instead of plain CNN, and our proposed CRCNet is one of the potential effective architectures.

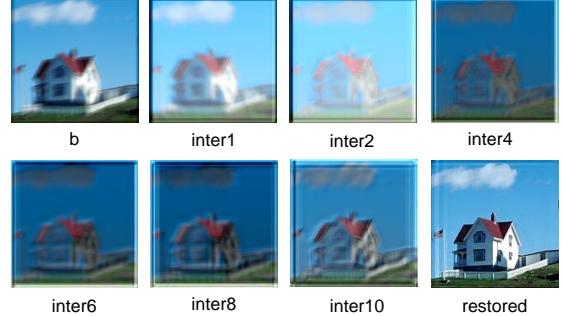


Figure 11: Intermediate samples of CRCNet. For visualization, deblurring is operated on the first channel of YCbCr and pixel values are normalized such that the maximum equals 1.

Conclusions

In this paper, we proposed an effective deep architecture for deconvolution. By deriving the MMSE-based deconvolution solution, we first proposed an iterative residual deconvolution algorithm, which is simple yet effective. We further designed a concatenated residual convolutional network with the basic architecture of IRD algorithm. The restored results by CRCNet are more visually plausible compared with competing algorithms. The success of CRCNet shows that deep CNN-based restoration architecture should borrow ideas from conventional methods. In the future, we will develop more effective deep CNN-based restoration methods for other low level vision tasks.

References

- Andrews, H. C., and Hunt, B. R. 1977. *Digital image restoration*. Englewood Cliffs, NJ: Prentice-Hall. chapter 7.2, 132–140.
- Chakrabarti, A. 2016. A neural approach to blind motion deblurring. In *ECCV*, 221–235. Springer.
- Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2014. Learning a deep convolutional network for image super-resolution. In *ECCV*, 184–199. Springer.
- Dong, W.; Shi, G.; and Li, X. 2013. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *IEEE Transactions on Image Processing* 22(2):700–711.
- Fergus, R.; Singh, B.; Hertzmann, A.; Roweis, S. T.; and Freeman, W. T. 2006. Removing camera shake from a single photograph. In *ACM Transactions on Graphics*, volume 25, 787–794. ACM.
- Girshick, R. 2015. Fast r-cnn. In *ICCV*, 1440–1448. IEEE.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 1026–1034.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778. IEEE.
- Hu, Z.; Huang, J.-B.; and Yang, M.-H. 2010. Single image deblurring with adaptive dictionary learning. In *ICIP*, 1169–1172. IEEE.
- Hu, W.; Xue, J.; and Zheng, N. 2012. Psf estimation via gradient domain correlation. *IEEE Transactions on Image Processing* 21(1):386–392.
- Jancsary, J.; Nowozin, S.; Sharp, T.; and Rother, C. 2012. Regression tree fields—an efficient, non-parametric approach to image labeling problems. In *CVPR*, 2376–2383. IEEE.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Krishnan, D., and Fergus, R. 2009. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 1033–1041. MIT Press.
- Kruse, J.; Rother, C.; and Schmidt, U. 2017. Learning to push the limits of efficient fft-based image deconvolution. In *ICCV*, 4596–4604. IEEE.
- Kundur, D., and Hatzinakos, D. 1996. Blind image deconvolution. *IEEE Signal Processing Magazine* 13(3):43–64.
- Kupyn, O.; Budzan, V.; Mykhailych, M.; Mishkin, D.; and Matas, J. 2018. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *CVPR*, 8183–8192. IEEE.
- Levin, A.; Fergus, R.; Durand, F.; and Freeman, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics* 26(3):70.
- Levin, A.; Weiss, Y.; Durand, F.; and Freeman, W. T. 2009. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 1964–1971. IEEE.
- Li, L.; Pan, J.; Lai, W.-S.; Gao, C.; Sang, N.; and Yang, M.-H. 2018. Learning a discriminative prior for blind image deblurring. In *CVPR*, 6616–6625. IEEE.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.
- Ma, C.; Yang, C.-Y.; Yang, X.; and Yang, M.-H. 2017. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding* 158:1–16.
- Martin, D.; Fowlkes, C.; Tal, D.; and Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, 416–423. IEEE.
- Nah, S.; Hyun Kim, T.; and Mu Lee, K. 2017. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 3883–3891. IEEE.
- Perrone, D., and Favaro, P. 2014. Total variation blind deconvolution: The devil is in the details. In *CVPR*, 2909–2916. IEEE.
- Ren, W.; Cao, X.; Pan, J.; Guo, X.; Zuo, W.; and Yang, M.-H. 2016. Image deblurring via enhanced low-rank prior. *IEEE Transactions on Image Processing* 25(7):3426–3437.
- Ren, D.; Zuo, W.; Zhang, D.; Xu, J.; and Zhang, L. 2018. Partial deconvolution with inaccurate blur kernel. *IEEE Transactions on Image Processing* 27(1):511 – 524.
- Schmidt, U., and Roth, S. 2014. Shrinkage fields for effective image restoration. In *CVPR*, 2774–2781. IEEE.
- Schmidt, U.; Rother, C.; Nowozin, S.; Jancsary, J.; and Roth, S. 2013. Discriminative non-blind deblurring. In *CVPR*, 604–611. IEEE.
- Tao, X.; Gao, H.; Shen, X.; Wang, J.; and Jia, J. 2018. Scale-recurrent network for deep image deblurring. In *CVPR*, 8174–8182. IEEE.
- Wang, Y.; Yang, J.; Yin, W.; and Zhang, Y. 2008. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences* 1(3):248–272.
- Xu, L.; Ren, J. S.; Liu, C.; and Jia, J. 2014. Deep convolutional neural network for image deconvolution. In *NIPS*, 1790–1798. MIT Press.
- Xu, L.; Tao, X.; and Jia, J. 2014. Inverse kernels for fast spatial deconvolution. In *ECCV*, 33–48. Springer.
- Yang, C.; Lu, X.; Lin, Z.; Shechtman, E.; Wang, O.; and Li, H. 2017. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, volume 1, 3. IEEE.
- Zhang, X.; Burger, M.; Bresson, X.; and Osher, S. 2010. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences* 3(3):253–276.
- Zhang, K.; Zuo, W.; Gu, S.; and Zhang, L. 2017. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 3929–3938. IEEE.
- Zuo, W.; Ren, D.; Gu, S.; Lin, L.; and Zhang, L. 2015. Discriminative learning of iteration-wise priors for blind deconvolution. In *CVPR*, 3232–3240.