

# Non-blind Deblurring: Handling Kernel Uncertainty with CNNs

Subeesh Vasu<sup>1</sup>, Venkatesh Reddy Maligireddy<sup>2</sup>, A. N. Rajagopalan<sup>3</sup>  
Indian Institute of Technology Madras

subeeshvasu@gmail.com<sup>1</sup>, venkateshmaligireddy@gmail.com<sup>2</sup> raju@ee.iitm.ac.in<sup>3</sup>

## Abstract

*Blind motion deblurring methods are primarily responsible for recovering an accurate estimate of the blur kernel. Non-blind deblurring (NBD) methods, on the other hand, attempt to faithfully restore the original image, given the blur estimate. However, NBD is quite susceptible to errors in blur kernel. In this work, we present a convolutional neural network-based approach to handle kernel uncertainty in non-blind motion deblurring. We provide multiple latent image estimates corresponding to different prior strengths obtained from a given blurry observation in order to exploit the complementarity of these inputs for improved learning. To generalize the performance to tackle arbitrary kernel noise, we train our network with a large number of real and synthetic noisy blur kernels. Our network mitigates the effects of kernel noise so as to yield detail-preserving and artifact-free restoration. Our quantitative and qualitative evaluations on benchmark datasets demonstrate that the proposed method delivers state-of-the-art results. To further underscore the benefits that can be achieved from our network, we propose two adaptations of our method to improve kernel estimates, and image deblurring quality, respectively.*

## 1. Introduction

Motion blur is a common and unpleasant corruption that occurs in hand-held photography and is difficult to undo. Blind image deblurring methods aim to recover the blur kernel as well as the clean image. Since the dimension of the kernel is much smaller than the image size, one can better constrain the estimation of the blur kernel rather than the image [21]. Hence, most existing blind deblurring (BD) approaches [10, 5, 38, 17, 40, 32, 35, 25] try to recover an accurate motion estimate from the blurred image [19]. This is eventually used to recover the latent image using an off-the-shelf non-blind deblurring (NBD) method.

The objective of NBD is to recover a sharp latent image from a known blurred image and a given blur estimate. Over the last decade, there has been significant progress

in this direction. Recent works [21, 16, 44] have come up with new image priors that can model the statistics of natural images which helps in suppressing ringing artifacts. Few works have even attempted to handle outliers such as noise [6, 7, 39, 14] and saturated regions [6, 39] to improve the quality of the deblurred result. Most of the existing non-blind methods are tailored to perform well under the assumption that the motion is known *accurately*. Consequently, they tend to underperform when the motion estimate is noisy. Because of the highly ill-posed nature of BD, an approach that can deliver accurate (close to ground truth) motion estimate is still far from reality [19]. The blur estimates from BD come with varying degrees of noise which can introduce unacceptable artifacts in the restored output.

In this paper, our focus is on improving restoration quality in the presence of noisy blur kernel. The approach that we propose consists of a conventional non-blind deblurring unit followed by a deep convolutional neural network (CNN) to remove undesired artifacts caused by errors in motion estimate. Recent works on single image restoration ([8, 9, 39]) have already revealed the potential of CNN based feature learning. The fact that we need discriminatory feature learning from multiple inputs renders CNN as a natural choice. Our approach is based on the premise that deblurred images obtained with different prior strengths carry complementary information. The complementarity lies in the fact that restored images with low prior weight preserve details but suffer from artifacts. On the other hand, a large prior weight helps in artifact removal but at the cost of image details. Hence, we are motivated to use multiple images obtained from the same blurred image-kernel pair, but with different prior strengths as inputs to our network. This we believe allows the network to perform better feature discrimination and restoration as compared to the single input image case. To handle arbitrary kernel noise, we also propose an approach to generate synthetic noisy kernels that can closely mimic the behavior of noise in real kernel estimates. We employ thousands of such synthetic noise kernels to improve the performance of our network and its generalization capability. Finally, as natural benefits that stem from our framework we advocate two adaptations

to improve the restoration quality further. We showcase the potential of our network to: i) reduce residual noise present in kernel estimate and ii) improve the performance of an existing BD algorithm.

Our main contributions are summarized below.

- We analyze the negative impact of residual noise in blur kernel estimates on image restoration quality, and propose an elegant solution based on CNN.
- We propose a systematic approach to synthetically generate noisy blur kernels that can closely mimic the behavior of real kernel estimates while simultaneously circumventing alignment issues associated with the training.
- We train our CNN with large numbers of synthetic and real noisy kernels to achieve state-of-the-art performance in non-blind deblurring.
- We propose two different adaptations of our trained network, both of which are intended to reduce the noise (which in turn results in better deblurring quality) associated with the kernel estimates from existing BD methods.

## 2. Related works

Classical methods such as Wiener filter [36] and Richardson-Lucy deconvolution [27] are known to cause ringing artifacts. Most of the works on NBD resort to maximum a posteriori (MAP) estimation, with differences in the type of the image prior they employ. While most existing works use global image priors [20, 33, 16] (typically in the form of  $\|\nabla l\|^\alpha$ , where  $\nabla l$  represents image gradient), the use of local (patch-based) priors [44] has been more effective in non-blind deconvolution. The work in [33] has shown that Laplacian prior ( $\alpha = 1$ ) is more effective than a Gaussian prior ( $\alpha = 2$ ), and can produce good results in reasonable time. However, according to recent studies [20, 16], the gradients of natural images are actually well-modeled by a hyper-Laplacian [20, 16] ( $0.5 \leq \alpha \leq 0.8$ ). The work in [44] is based on expected patch log likelihood (EPLL), a form of patch prior learned from natural images to yield state-of-the-art results.

Few other works have tried to handle the effects of outliers. The works in [6, 35] account for the presence of saturation in images. Works in [7, 14] have tried to remove deconvolution artifacts introduced by image noise, but the restored results are significantly affected by ringing artifacts induced by kernel errors. [13] proposes a new regularization method for removing artifacts produced by kernel errors but fails to retain high-frequency details.

Recent works using deep neural networks have shown promising results. [39] trained a deep CNN to perform inversion of a single kernel in the presence of non-linearities in the blurred image. The work in [30] has used multi-layer perceptron to perform outlier-robust image restoration by learning to predict the latent image from an initial estimate obtained using  $L_2$  norm prior. But the scope

of their network is limited to a single specific blur kernel. Some of the recent works on NBD employ machine learning frameworks such as Gaussian conditional random fields [29], shrinkage fields [28], whereas the most recent work in [18] uses CNN based regularization. However, none of these methods can handle noisy blur kernels.

## 3. Non-blind deblurring: Role of image prior

In a convolutional model, the latent image  $l$  is related to the blurred image  $b$  through blur kernel  $k$  as

$$b = l * k + n \quad (1)$$

where ‘\*’ refers to the convolution operation and  $n$  is Additive White Gaussian (AWG) noise. Most of the existing NBD methods adopt a MAP formulation, where an estimate of the latent image is obtained from kernel estimate  $\hat{k}$  by solving an optimization framework of the form

$$\hat{l} = \arg \min_l \frac{\lambda}{2} \|l * \hat{k} - b\|^2 + P(l) \quad (2)$$

where  $P$  determines prior on the latent image, and  $\lambda$  is the weight of the data cost over the prior term. One of the most commonly adopted approaches is to enforce global priors as

$$P(l) = \|\nabla l\|^\alpha \quad (3)$$

where  $\nabla l$  represents image gradient. Recent studies shows that the distribution of natural image gradients can be well-modeled by a hyper-Laplacian [20, 16] ( $0.5 \leq \alpha \leq 0.8$ ). Apart from the use of global image priors, the state-of-the-art work in [44] uses a local patch based prior given by

$$P(l) = \sum_i \log p(E_i l) \quad (4)$$

where  $E_i$  is a matrix which extracts the  $i^{th}$  patch from the image, and  $\log p(E_i l)$  is the likelihood of the  $i^{th}$  patch under the prior defined by  $p$ .

Our proposed approach is built on the observation that irrespective of the nature of prior employed in Eq. 2,  $\lambda$  has a crucial role to play in the final image restoration quality.

### 3.1. Prior weight and restoration quality

In this section, we will discuss the impact on the restoration results from Eq. 2 with respect to variations in  $\lambda$ . For noisy blur kernels, existing prior based NBD works deliver artifact-free images by keeping the prior weight high (*i.e.*, a low  $\lambda$  in Eq. 2). However, this comes at the cost of loss of details. To illustrate this impact, we will consider the cases of 3 different priors: Eq. 3 with  $\alpha = 2$  and  $\alpha = 2/3$ , and Eq. 4. To differentiate, we denote the weight of the data cost for each method as  $\lambda_2$ ,  $\lambda_{2/3}$ , and  $\lambda_p$ , respectively. Fig. 1 illustrates the differences in the restored images while we

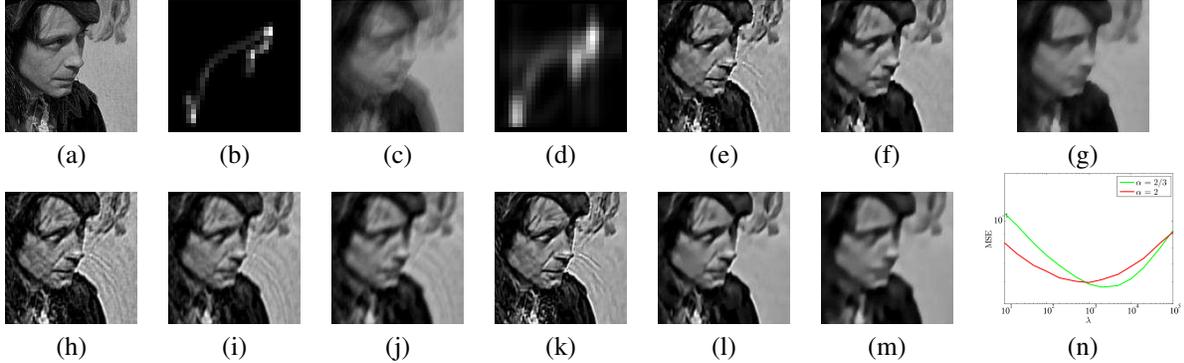


Figure 1. Impact on the restored image quality with varying  $\lambda$ . (a) Latent image. (b) Ground truth kernel. (c) Blurred image generated using (a) and (b). (d) Noisy kernel obtained from [32]. Estimated images using kernel in (d) with  $\lambda_{2/3}$  = (e)  $2e4$ , (f)  $2e3$ , and (g)  $2e2$ , with  $\lambda_2$  = (h)  $1e4$ , (i)  $1e3$ , and (j)  $1e2$ , and with  $\lambda_p$  = (k)  $64e5$ , (l)  $64e4$ , and (m)  $64e3$ . (n) Average MSE of restored images for varying  $\lambda$  values.

vary the value of  $\lambda$ . The restored images using the noisy kernel in Fig. 1(d) corresponding to the three NBD methods are shown in Figs. 1(e-m). For all the methods, while a higher value of  $\lambda$  results in more details, it also introduces artifacts. Decreasing  $\lambda$  increases the influence of the prior term, reduces ringing artifacts but results in loss of high-frequency details. When the prior weight goes high, while Gaussian prior (Figs. 1 (f-g)) results in blurry edges,  $\alpha = 2/3$  and [44] preserve sharpness of strong gradients. However, loss of details is a common issue with all prior-based methods. This is due to the fundamental limitation of image priors in representing the contents of natural images in their full generality. Often, this leads to partial restoration *i.e.*, the restored contents will be close to the ground truth only for those areas which respect the prior distribution.

## 4. Proposed approach

In this paper, we propose an elegant approach for detail-preserving restoration of blurry images under the assumption of a noisy blur kernel. Our approach for image restoration consists of two modules: a conventional NBD unit to obtain multiple initial estimates of the latent image followed by the use of a deep CNN to remove any undesired artifacts present in the initial estimates, and to provide enhanced details. We use multiple initial estimates as inputs to the network since the complementarity among the inputs leads to improved restoration performance. In this paper, our primary focus is on training the CNN by initializing from restored results of [16] (*i.e.*, Eq. 3 with  $\alpha = 2/3$ ). We use [16] for initialization since among the works that employ global priors, [16] gives the best restoration results, and as compared to the state-of-the-art work of [44], the performance of [16] is the closest (among all existing prior-driven NBD methods) while being much faster. For the hyper-Laplacian prior, we use  $\alpha = 2/3$ , the value at which the restoration quality was found to be the best as reported in [16]. Our approach can be applied to initial estimates from

other weight dependent prior based NBD approaches too. To assess the performance improvement when our approach is used in conjunction with other NBD methods, we also include analysis for the case when the inputs to our network come from a Gaussian prior-based NBD approach (*i.e.*, Eq. 3 with  $\alpha = 2$ ).

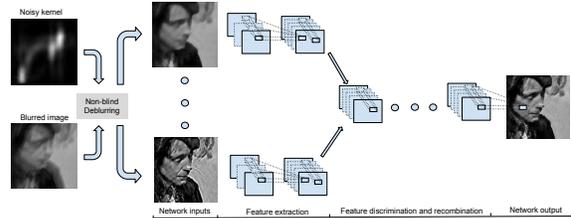


Figure 2. Our network structure. We use the deblurred images obtained from a standard NBD scheme corresponding to different prior strengths as inputs to our FCN.

### 4.1. Motivation

Our idea of using deep CNN to improve the output from a deconvolution unit is motivated by the fact that  $\lambda$  plays a crucial role in deciding the restoration quality. As depicted in Fig. 2, instead of giving the blurry observation, we provide as input the restored images obtained from an existing NBD approach. The network is then trained to learn kernel-independent features. However, this alone does not solve the problem entirely. This is because, unlike the type of artifacts addressed in previous works on denoising [3, 37] or dirt removal [9], the variability of the artifacts introduced by blur kernel noise is much more complicated. Hence, direct use of deep networks designed for afore-mentioned applications does not work for our scenario.

To analyze this point further, let us consider different cases that are possible in our scenario. If we use a single restored image (refer to Figs. 1(g,j,m)) obtained using low  $\lambda$  alone as input, the role of the network will be to restore the lost details. While this appears similar to single image

super-resolution [8], ours is a more difficult task since the degradation is relatively complicated for NBD. This is also reflected in our attempts to improve restoration by training on such inputs, which did not lead to any significant improvements in the restoration quality. On the other hand, we observed that using as input the restored image obtained with high  $\lambda$  leads to better restoration. This is because the objective of the network is then limited to removing undesirable artifacts present in the image. Our key finding is that the restored outputs corresponding to low and high values of  $\lambda$  contain complimentary information.

## 4.2. Network architecture

The proposed network structure for NBD is shown in Fig. 2. First, we use the blurry image and noisy kernel estimate to generate multiple estimates (corresponding to different values of  $\lambda$ ) of the latent image using a conventional NBD scheme. These multiple image estimates are passed through individual feature extraction units formed from two convolutional layers. The extracted features are then combined and passed through a number of convolutional layers which act as a feature discrimination unit using which the desired artifact-free features are integrated to yield the final restored image. Since CNN can learn feature discrimination as well as patch specific priors from the training data, our approach also attempts to restore enhanced details in output.

The feature extraction unit corresponding to each input consists of two convolutional layers, with 64 and 128 filters in the first and second layer, respectively. Thus when we train with  $m$  inputs, the feature discrimination unit takes  $128m$  feature maps from the feature extraction unit as input. Our feature discrimination unit consists of 7 convolutional layers with the number of filters in each layer being 512, 512, 512, 512, 128, 64, and 1, respectively. At the output of every convolution layer, we apply batch-normalization followed by ReLu [11]. For all the layers, we use filters of size  $3 \times 3$  with a stride of 1 and zero-padding by 1, to maintain spatial resolution over the entire network. The hyperparameter settings that we use are partially motivated from the encoder-decoder architecture used in [12], although our network has significant differences in terms of structure. A detailed discussion on the various architectures that were attempted has been provided in the supplementary material.

## 5. Training

We use  $L_2$  norm of the difference between the network output and the ground truth sharp image to define the loss for training. For optimization, we used ADAM [15] with learning rate 0.0002 and momentum 0.5.

### 5.1. Data generation

As we have already discussed, we use the images obtained with different values of  $\lambda$  as inputs to our network.

Images from the BSD 500 dataset [1] are used as latent images for generation of training and test data. While the first 400 images were used for training, the remaining 100 images were used for testing. In order to learn a network which can generalize well to arbitrary noises in the kernels, we need to use a large number of realistic noisy kernels to generate the training data. As noisy kernel estimates, we used 3200 kernels returned by 5 BD methods ([17, 22, 32, 4], [32] has 2 BD methods) while deblurring 640 blurry images (formed using 80 sharp images and 8 ground truth (GT) kernels) from [32]. Of these, we use 1.6K from the first 4 GT kernels for training and the remaining for testing (mutually exclusive). We followed patch-wise training in which randomly cropped patches of size  $101 \times 101$  from the estimated images were used as inputs to our network. More details on our training is provided in the supplementary material.

## 6. Optimal input identification

In order to identify the input images that can deliver best restoration quality, we trained our network with different input combinations. We began our experiments by training with single image inputs and then proceeded to add more inputs to the network. To find the set of input parameters that can be used for this analysis, we quantitatively measured the restoration accuracy of a set of test images for different values of  $\lambda$ . We used the images from set14 dataset of [41] as latent images, and eight ground truth kernels from [22] to generate 100 blurred images. These images were then restored using kernel estimates returned by the BD methods in [10] and [22] to obtain a set of 200 restored images corresponding to  $\alpha = 2$  as well as  $\alpha = 2/3$ , while varying  $\lambda$  over a suitable range. For each value of  $\lambda$ , we compute the average mean squared error (MSE) for all the restored images. A plot of the average MSE over this test set is shown in Fig. 1(n). As expected, for both methods the MSE achieves a minimum value at an intermediate value of  $\lambda$  and gradually increases as  $\lambda$  starts to move away from this point.

Based on visual inspection of image quality and our quantitative evaluation in Fig. 1(n), we chose  $\lambda_{2/3} = 2e4$ ,  $2e3$ , and  $2e2$  to generate multiple inputs corresponding to [16]. This is because: (i) [16] has shown that  $\lambda_{2/3} = 2e3$  yields optimal restoration quality, which is also supported by the results from our experiments (as in Fig. 1(n)), and (ii) the restored results corresponding to  $\lambda_{2/3} = 2e2$  and  $2e4$  had significant quantitative and qualitative differences with respect to the optimal  $\lambda_{2/3}$  which is  $2e3$ .

Fig. 3 illustrates differences in the restoration quality when we train the network with the best performing 1, 2, and 3 input cases. Fig. 4 displays the average peak signal-to-noise ratio (PSNR) value (averaged over all images in the test data) of the network outputs while training with different input combinations. Out of all possible single-input cases, the output of network trained with input correspond-

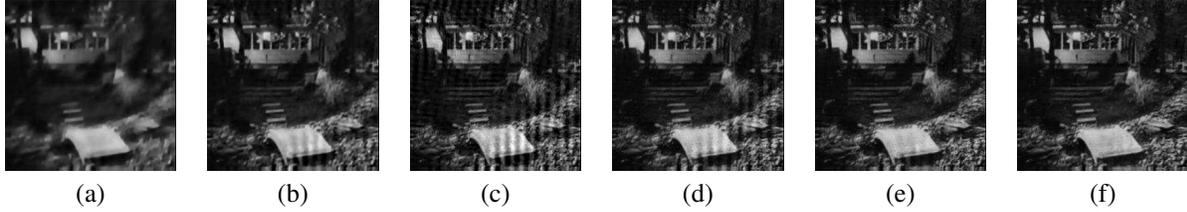


Figure 3. Performance of our network with different input combinations. Inputs to the network with (a)  $\lambda_{2/3} = 2e2$ , (b)  $\lambda_{2/3} = 2e3$ , and (c)  $\lambda_{2/3} = 2e4$ . Output of the network trained using (d)  $\lambda_{2/3} = 2e4$ , (e)  $\lambda_{2/3} = 2e3, 2e4$ , and (f)  $\lambda_{2/3} = 2e2, 2e3, 2e4$ .

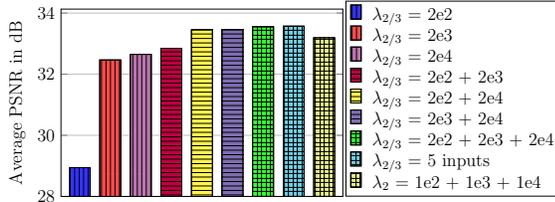


Figure 4. Network performance as a function of inputs.

ing to  $\lambda_{2/3} = 2e4$  yielded the highest PSNR. Among all the input pairs, it was found that  $2e2 + 2e3$  performed the worst (mainly due to lack of details in both the inputs) whereas  $2e3 + 2e4$  performed the best with a marginal improvement over  $2e2 + 2e4$ . When we use a single input corresponding to  $\lambda_{2/3} = 2e4$ , the network was able to achieve only partial artifact removal (Fig. 3(d)). Whereas with 2 inputs ( $2e3 + 2e4$ ) most of the artifacts went away (Fig. 3(e)). While the PSNR improvement for the 2-input case was significant, this was not true when we moved from 2 to 3 inputs. Fig. 3(f) reveals the importance of the third input ( $\lambda_{2/3} = 2e2$ ) in removing artifacts in Fig. 3(e). We also conducted experiments by adding more inputs to the network. However, it was observed that further addition of inputs yields only a marginal improvement in PSNR. This is evident from Fig. 4, where the network output PSNR corresponding to 5 inputs ( $2e2 + 5e2 + 2e3 + 5e3 + 2e4$ ) can be observed to be very close to that of the 3-input case. Among all possible input combinations, the 3-input network was observed to yield best performance overall (Fig. 3(f)) and was hence adopted for analyzing our network further.

Our experiments to identify the optimal inputs for  $\alpha = 2$  also lead to very similar observations. The best 3 input combination for this case turned out to be  $\lambda_2 = 1e4, 1e3$ , and  $1e2$ . While the overall performance improvement over the inputs was quite good for the case of  $\alpha = 2$ , the results were marginally inferior (refer Fig. 4) as compared to that of  $\alpha = 2/3$ . Therefore, the performance of the network output depends on the initialization, and hence there exists scope to further improve the performance by adopting NBD approaches with better priors. This is an added advantage since the performance of our approach scales up along with improvements in image-prior-driven restoration schemes. Note that while adding further combinations by

changing both  $\lambda$  and  $\alpha$  to produce inputs is certainly possible, we rely on the principle assumption that regions with fewer artifacts and high-frequency details must go as inputs to the network. This is satisfactorily achieved by keeping  $\alpha$  constant and changing only  $\lambda$ .

## 7. Synthetic vs real noisy kernel

To obtain real noisy kernels, one can use the blur kernels returned by conventional BD algorithms. However, this has several issues. First of all, the centroid of kernel estimates returned by BD algorithms may not be consistent with the centroid of the ground truth kernels. This will introduce random misalignments between the deblurred image and corresponding latent image, thus hindering the learning capability of the network. Secondly, it is impossible to obtain a large number of real noisy kernels since the generation of real kernels is very time-consuming. To address these issues, we attempt two different strategies to use noisy kernels for training. Our first approach is to use the real kernels for training, and account for any misalignments introduced by the kernel noise by performing affine registration [2] of the estimated image with the ground truth image. The second is to generate synthetic noisy kernels.

### 7.1. Generation of synthetic noisy kernels

We used an optimization framework to generate synthetic noisy kernels. To make the network generalize well-enough to handle real kernels, the generated synthetic noisy kernels must mimic the characteristics of real noise encountered in practice. Typical kernel priors used by BD methods ( $L_2$  or  $L_1$  norm on kernels) tend to deliver smoothly varying kernel estimates, while suppressing isolated noises which might get generated otherwise. We have tried to analytically model and adhere to these properties in our synthetic kernel generation scheme. To resolve the issue with respect to changes in the centroid between ground truth and noisy kernels, we perform centroid-preserving noise addition *i.e.*, we first align the ground truth kernels ( $\mathbf{k}_{gt}$ ) such that their centroid is zero. Then, we generate zero-centroid synthetic kernel noise ( $\hat{\mathbf{n}}$ ) using our optimization framework (discussed next) and add it to the ground truth kernel to yield the desired noisy kernel. This ensures that both the restored image and the ground truth image are perfectly aligned.

To generate  $\hat{\mathbf{n}}$ , we solve the optimization problem

$$\hat{\mathbf{n}} = \arg \min_{\mathbf{n}} \{ \|\mathbf{n} - \lambda_1 \mathbf{W}_v \mathbf{n}_i\|^2 + \lambda_c \|\mathbf{M}_c \mathbf{W}_v \mathbf{n}\|^2 + \lambda_p (\|\mathbf{W}_v \mathbf{P}_x \mathbf{n}\|^2 + \|\mathbf{W}_v \mathbf{P}_y \mathbf{n}\|^2) \} \quad (5)$$

Here,  $\mathbf{n}_i$  is an initial estimate of kernel noise which we obtain using a random number generator. While  $\lambda_1$  determines the depth of the noise,  $\mathbf{W}_v$  is the mask for selecting only the valid region over the neighborhood of the ground truth kernel.  $\mathbf{M}_c$  is the centroid computation matrix, and  $\lambda_c$  is the weight for centroid enforcement term.  $\mathbf{P}_x$  and  $\mathbf{P}_y$  are matrices for gradient computation along  $x$  and  $y$ , and  $\lambda_p$  is the smoothness weight over the gradients of kernel noise. We also enforce an additional non-negativity constraint on kernel noise. In Eq. 5, the first term ensures randomness of the generated noise by enforcing it to be closer to  $\mathbf{n}_i$ . The final synthetic noisy kernel  $\hat{\mathbf{k}}_s$  is formed by adding  $\hat{\mathbf{n}}$  to the ground truth kernel followed by Gaussian low-pass filtering (with variance  $v_g$ ) as

$$\hat{\mathbf{k}}_s = \mathbf{f}_{v_g} * (\mathbf{k}_{gt} + \hat{\mathbf{n}}) \quad (6)$$

Low-pass filtering enforces the desired smoothing effect. We chose  $\lambda_c = 1e3$ , and  $\lambda_p = 1$  for all our experiments. For training, we used synthetic noisy kernels obtained with  $6 \leq \lambda_1 \leq 9$  and  $0.5 \leq v_g \leq 0.7$ . To form the training data corresponding to synthetic noisy kernels, we used the approach in [4] to generate ground truth kernels.<sup>1</sup>

The average PSNR gain (in dB) obtained on our test data, with networks trained using real and synthetic kernels individually was found to be 0.89 and 0.91. Although the synthetic noisy kernels were obtained from a larger variety of GT kernels as compared to real kernels, the PSNR gain using only the synthetic noisy kernels was just comparable to that of real kernels. This is probably because the noisy kernels which we generated despite being visually similar to the real kernels, may not represent the behavior of real kernels in their complete generality. However, when we used synthetic kernels and real kernels together for training, the PSNR gain improved to 1.15. Our proposed kernel generation scheme delivers a large number of synthetic kernels quickly allowing us to incorporate a larger set of noisy kernels generated by a wide variety of GT kernels into our training process. Since the collections of real and synthetic kernels have differences in terms of variations in kernel noise and nature of GT kernels, combined training enabled the network to generalize better. To the best of our knowledge, this is the first attempt of its kind to systematically generate noisy blur kernels, with *implicit alignment*. Potential exists to harness this capability in a variety of related image restoration tasks such as blind deblurring.

<sup>1</sup>An illustration of the closeness between real noisy kernels and our synthetically generated kernels is provided in the supplementary material.

## 8. Results and comparisons

In this section, we evaluate the performance of our network by comparing with the state-of-the-art works in the literature. For all the experimental results presented here, we use a network trained with synthetic and real, as well as low and high-noise kernels. To evaluate the performance of our proposed method, we use the publicly available datasets in [22, 32, 19].

Table 1. Average PSNR on dataset of [22]

NBD method	Method for kernel estimation			
	[10]	[5]	[22]	[5]+refinement
[13]	27.28	28.55	28.79	28.92
[14]	27.12	28.07	28.11	28.78
[7]	28.61	28.84	29.20	29.45
[18]	28.74	28.99	29.51	29.75
[20]	28.86	29.32	29.49	29.63
[16]	29.08	29.39	29.55	29.81
[28]	29.29	29.50	29.66	29.90
[44]	29.51	29.66	29.85	30.01
CNN <sub>2/3</sub> (1)	29.50	29.64	29.91	30.22
CNN <sub>2/3</sub> (3)	<b>30.40</b>	<b>30.62</b>	<b>30.87</b>	<b>31.19</b>
CNN <sub>2</sub> (3)	29.91	30.25	30.30	30.69

The most desirable property of an NBD algorithm is to maintain good restoration quality for kernel estimates returned by arbitrary BD methods. To verify this generalization capability of our method, we perform extensive quantitative and qualitative evaluation using real kernels returned by several BD methods. For a particular BD method, we use the corresponding set of kernels to obtain the restored images using different NBD methods. The quality of these deblurred images is used to assess performance. For the dataset in [22], we used kernel estimates obtained from [10, 5, 22], whereas for [32] we used kernel estimates returned by [5, 38, 23]. Since [19] comprises of many challenging examples, there exist examples on which BD algorithms ([38, 40, 32, 26]) fail to recover the blur kernels. To avoid the damaging influence of such bad kernel estimates, we excluded them while testing performance in Table 2.

To evaluate the performance of our proposed approach, we compared with existing NBD approaches in [20, 16, 44, 13, 7, 28, 14, 18] and used PSNR, SSIM ([34]), and IFC [31] as metrics. We used online-available implementation of these NBD methods with optimal parameters settings as specified in the respective works. For [16], we used  $\lambda = 2e3$ , as given therein. Use of IFC for [19] is motivated by the fact that on the images from this dataset, IFC has the highest correlation with respect to human subject scores [19]. As can be observed from Tables 1 and 2, our method significantly outperforms the state-of-the-art works with respect to *all* the metrics. In Table 1, for our proposed method,

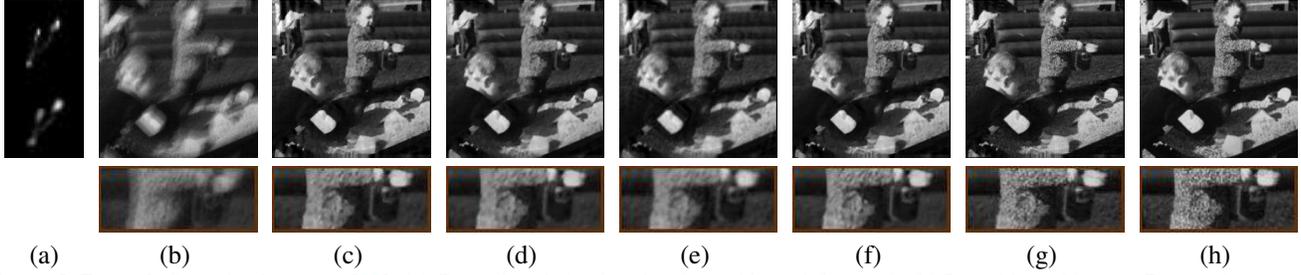


Figure 5. Example from the dataset in [22]. (a) Ground truth (top) and estimated kernel (bottom). (b) Input blurred image. Restored images using (c) [16] (d) [20], (e) [28], (f) [44], and (g) proposed approach ( $\text{CNN}_{2/3}$ ). (h) GT image.

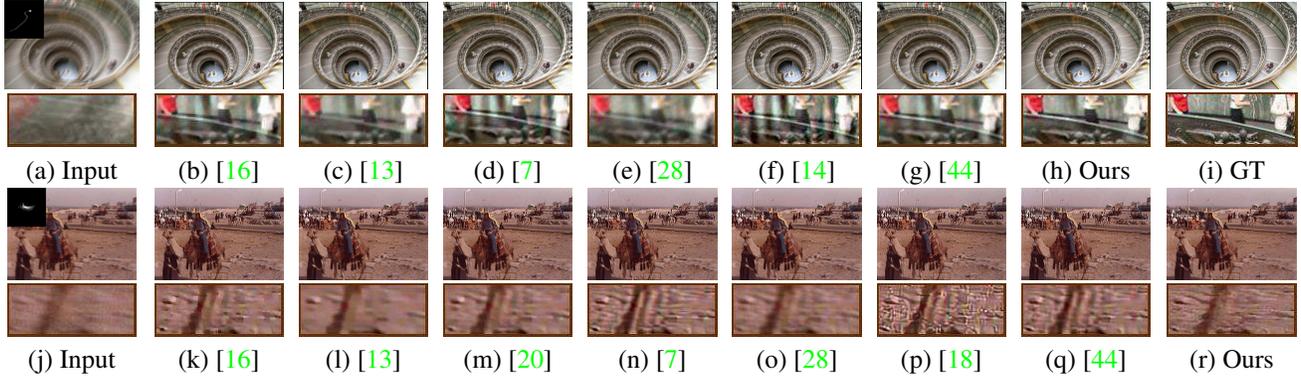


Figure 6. Synthetic (Row 1) and real (Row 2) examples from the dataset in [19].

the number in parentheses indicates the number of inputs used for CNN. While with a single input (Table 1,  $\text{CNN}_{2/3}$  (1)), the performance of our method is only comparable to other works, the addition of more inputs results in significant improvement in PSNR. Furthermore, although the performance of  $\text{CNN}_2$  is better than the competing methods,  $\text{CNN}_{2/3}$  performs the best.

For visual comparisons, we show few randomly chosen representative examples.<sup>2</sup> Fig. 5 shows a low resolution example from [22]. We used the kernel estimate shown in Fig. 5(a) to deblur the image in Fig. 5(b). As is evident from Fig. 5(g), our approach restores the image without artifacts while achieving significant improvement in the recovery of details over competing methods. Fig. 6 reveals performance on high resolution images from [19]. While our approach (Figs. 6(h,r)) is able to deliver an artifact-free and yet detail-preserving image, the competing methods (Figs. 6(b-g,k-q)) either fail to recover the details or generate significant artifacts. Similar observations can be found from our extensive comparisons on kernels from various BD methods ([10, 5, 38, 22, 17, 40, 32, 42, 43, 23, 26, 24, 25]) revealing the generalization capability of our trained network.

<sup>2</sup>Our supplementary material contains discussions on few other training experiments which we have conducted, analysis on generalization capability of our network to other NBD methods, run-time comparisons for the NBD methods, additional quantitative comparisons and extensive visual comparisons on more examples.

## 9. Other improvisations

In this section, we demonstrate that there are further benefits to be derived from our proposed framework. We first show how our NBD method can be employed in an iterative fashion to reduce the residual noise in the kernels and improve restoration quality further. Next, we reveal how our NBD method can be used as a plug-in inside an existing BD method to mitigate local minima issues which is at the core of erroneous blur kernel estimates.

### 9.1. Iterative restoration for kernel noise reduction

As discussed earlier, kernel estimate from BD methods is often affected by noise. The noise level specific training experiments that we have conducted with our network (details are provided in the supplementary material) reveal that the average PSNR is higher for test data corresponding to less-noisy kernels. While our approach performs significantly better than the existing NBD methods, the performance of our trained network is fundamentally limited by the noise level in the input kernel (akin to other NBD methods). We explored the possibility of reducing the kernel noise level by employing our NBD approach in an iterative fashion. More specifically, we treat the input kernel as the initial kernel estimate  $\hat{k}_1$  and solve the following optimization problem to obtain refined estimates of blur kernel.

$$\hat{k}^{t+1} = \arg \min_k \|(F \circ \nabla \hat{l}_{net}^t) * k - \nabla b\|^2 + \beta \|k\|^2 \quad (7)$$

Table 2. Performance comparison on dataset of [32] and [19]

NBD method	BD method for kernel estimation							
	PSNR/SSIM for images from [32]			SSIM/IFC for images from [19]				
	[5]	[38]	[23]	[38]	[40]	[32]	[26]	[40] + CNN <sub>2/3</sub>
[20]	28.02/0.81	29.56/0.83	28.87/0.80	0.72/2.17	0.72/2.00	0.69/1.92	0.69/1.91	0.75/2.31
[16]	28.29/0.83	30.15/0.85	29.27/0.82	0.73/2.40	0.73/2.16	0.71/2.08	0.71/2.13	0.75/2.50
[28]	27.22/0.78	28.34/0.81	27.90/0.78	0.66/1.70	0.67/1.60	0.65/1.60	0.63/1.50	0.70/1.88
[44]	28.46/0.83	30.51/0.86	29.59/0.82	0.75/2.52	0.74/2.28	0.72/2.20	0.72/2.20	0.76/2.59
CNN <sub>2/3</sub>	<b>29.54/0.88</b>	<b>32.60/0.91</b>	<b>30.69/0.85</b>	<b>0.78/2.58</b>	<b>0.76/2.30</b>	<b>0.75/2.31</b>	<b>0.75/2.41</b>	<b>0.80/2.81</b>

where  $t$  denotes the iteration number,  $\nabla \hat{l}_{net}^t$  is the gradient of the restored image obtained using our NBD method (Fig. 2) from the corresponding kernel estimate  $\hat{k}^t$ . We apply an additional bilateral filtering operation (denoted as  $F \circ$ ) to remove small noisy gradients present in  $\nabla \hat{l}_{net}^t$  that can be detrimental to kernel estimation. We iteratively repeat our network-based restoration (Fig. 2) and kernel estimation steps to yield improved restoration quality. We empirically found the number of required iterations for convergence of this alternating minimization to be about 5. We used  $\beta = 20$  in our experiments. The last column in Table 1 corresponds to quantitative evaluation of our proposed kernel refinement scheme, while attempting to reduce the noise levels from kernels returned by [5]. Clearly, the refined kernels from our approach result in significant performance improvement over the kernels returned by [5].

## 9.2. Plug-in for blind deblurring

Most existing BD approaches employ an alternating minimization scheme for kernel estimation where they alternately estimate image and blur kernel until convergence. The main difference between different BD methods lies in the prior employed for image and/or kernel estimation. One of the main problems associated with BD methods is that the alternating minimization scheme often gets stuck in local-minima due to continuous accumulation of kernel errors. Therefore, such BD methods (which employ a latent image estimation unit akin to ours) can harness the potential of our trained network to improve their performance. Since our approach can produce restored images devoid of artifacts, we propose to replace the latent image estimation unit of such BD methods with our NBD approach to alleviate local-minima issues associated with kernel errors.

To validate this possibility, we incorporated our NBD approach within the BD approach of [40]. To perform intermediate image estimation, [40] enforces  $L_0$  sparsity on the gradients of the image by solving the following set of equations.

$$\hat{l} = \arg \min_l \frac{\lambda}{2} \|l * \hat{k} - b\|^2 + \|\nabla l - z\|^\alpha \quad (8)$$

$$z = \begin{cases} 0, & \text{if } \nabla \hat{l} < \epsilon \\ \nabla \hat{l}, & \text{otherwise} \end{cases} \quad (9)$$

Both Eq. 8 and Eq. 9 are alternatively solved by varying  $\epsilon \in \{1, 2^{-1}, 4^{-1}, 8^{-1}\}$  from 1 to  $8^{-1}$ . The work in [40] uses  $\alpha = 2$  for optimization. We note that convergence directly depends on the initial estimate obtained for the case of  $\epsilon = 1$ . Furthermore, for  $\epsilon = 1$  both Eq. 8 and Eq. 9 are equivalent to Eq. 2, since  $z$  will be 0. Hence, as an improvisation, we replace the initial estimate obtained for the case of  $\epsilon = 1$  with our NBD approach corresponding to  $\alpha = 2/3$  to ensure better convergence. We name the corresponding BD approach as ‘[40] + CNN<sub>2/3</sub>’. Quantitative evaluation in Table 2 clearly reveals that our proposed BD approach ‘[40] + CNN<sub>2/3</sub>’ is able to deliver significant improvements in restoration as compared to [40] alone.<sup>3</sup>

## 10. Conclusions

We presented a deep CNN-based framework for non-blind restoration of motion blurred images. Unlike existing works, we investigated a very relevant scenario which is the unavailability of exact ground truth kernel. By using multiple latent image estimates obtained with different prior strengths as inputs, our network exploits the complementarity present in the input data to yield high-quality restoration results. To remove kernel noise-specific artifacts in the deconvolved results, we trained our network with real kernels obtained from existing blind deblurring methods as well as synthetically generated noisy kernels. Our method is able to deliver state-of-the-art performance in non-blind deblurring. Furthermore, as indicated by our proposed improvisations, one can further reduce the noise level in kernel estimates via our NBD approach and improve the overall restoration quality.

**Acknowledgements:** The first author thanks Sunil Kumar for his help in running some comparisons. The authors gratefully acknowledge the travel grant support from Google Research India.

<sup>3</sup>Qualitative comparisons on our improvisations are provided in the supplementary material.

## References

- [1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. 4
- [2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004. 5
- [3] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, pages 2392–2399. IEEE, 2012. 3
- [4] A. Chakrabarti. A neural approach to blind motion deblurring. In *ECCV*, pages 221–235. Springer, 2016. 4, 6
- [5] S. Cho and S. Lee. Fast motion deblurring. In *TOG*, volume 28, page 145. ACM, 2009. 1, 6, 7, 8
- [6] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. In *ICCV*, pages 495–502. IEEE, 2011. 1, 2
- [7] A. Danielyan, V. Katkovnik, and K. Egiazarian. Bm3d frames and variational image deblurring. *TIP*, 21(4):1715–1728, 2012. 1, 2, 6, 7
- [8] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2016. 1, 4
- [9] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, pages 633–640, 2013. 1, 3
- [10] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *TOG*, volume 25, pages 787–794. ACM, 2006. 1, 4, 6, 7
- [11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 4
- [12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016. 4
- [13] H. Ji and K. Wang. Robust image deblurring with an inaccurate blur kernel. *TIP*, 21(4):1624–1634, 2012. 2, 6, 7
- [14] A. Kheradmand and P. Milanfar. A general framework for regularized, similarity-based image restoration. *TIP*, 23(12):5136–5151, 2014. 1, 2, 6, 7
- [15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4
- [16] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, pages 1033–1041, 2009. 1, 2, 3, 4, 6, 7, 8
- [17] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, pages 233–240. IEEE, 2011. 1, 4, 7
- [18] J. Kruse, C. Rother, and U. Schmidt. Learning to push the limits of efficient fft-based image deconvolution. In *ICCV*, Oct 2017. 2, 6, 7
- [19] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang. A comparative study for single image blind deblurring. In *CVPR*, June 2016. 1, 6, 7, 8
- [20] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *TOG*, 26(3):70, 2007. 2, 6, 7, 8
- [21] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971. IEEE, 2009. 1
- [22] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664. IEEE, 2011. 4, 6, 7
- [23] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV*, pages 783–798. Springer, 2014. 6, 7, 8
- [24] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *CVPR*, pages 2901–2908, 2014. 7
- [25] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *CVPR*, June 2016. 1, 7
- [26] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, pages 2909–2916, 2014. 6, 7, 8
- [27] W. H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972. 2
- [28] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR*, pages 2774–2781, 2014. 2, 6, 7, 8
- [29] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative non-blind deblurring. In *CVPR*, pages 604–611, 2013. 2
- [30] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, pages 1067–1074, 2013. 2
- [31] H. R. Sheikh, A. C. Bovik, and G. De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *TIP*, 14(12):2117–2128, 2005. 6
- [32] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, pages 1–8. IEEE, 2013. 1, 3, 4, 6, 7, 8
- [33] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. 2
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004. 6
- [35] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. *IJCV*, 110(2):185–201, 2014. 1, 2
- [36] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*, volume 7. MIT press Cambridge, MA, 1949. 2
- [37] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *NIPS*, pages 341–349, 2012. 3
- [38] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, pages 157–170. Springer, 2010. 1, 6, 7, 8
- [39] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, pages 1790–1798, 2014. 1, 2

- [40] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, pages 1107–1114, 2013. [1](#), [6](#), [7](#), [8](#)
- [41] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. [4](#)
- [42] H. Zhang, D. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *CVPR*, pages 1051–1058, 2013. [7](#)
- [43] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *CVPR*, pages 612–619, 2013. [7](#)
- [44] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486. IEEE, 2011. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)