Motion Deblurring in the Wild

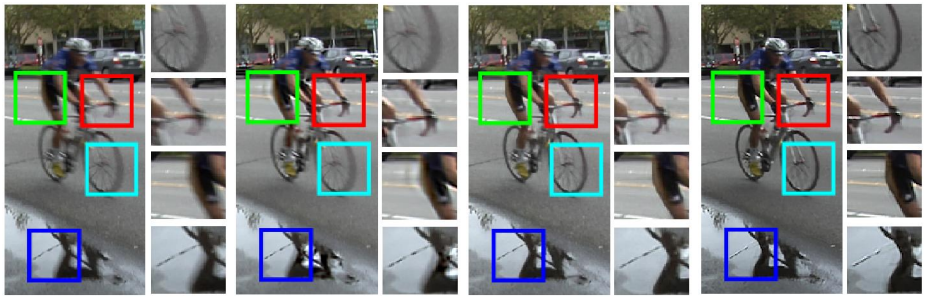
Mehdi Noroozi, Paramanand Chandramouli, Paolo Favaro

Institute for Informatics
University of Bern
{noroozi, chandra, paolo.favaro}@inf.unibe.ch

Abstract. We propose a deep learning approach to remove motion blur from a single image captured *in the wild*, *i.e.*, in an uncontrolled setting. Thus, we consider motion blur degradations that are due to both camera and object motion, and by occlusion and coming into view of objects. In this scenario, a model-based approach would require a very large set of parameters, whose fitting is a challenge on its own. Hence, we take a data-driven approach and design both a novel convolutional neural network architecture and a dataset for blurry images with ground truth. The network produces directly the sharp image as output and is built into three pyramid stages, which allow to remove blur gradually from a small amount, at the lowest scale, to the full amount, at the scale of the input image. To obtain corresponding blurry and sharp image pairs, we use videos from a high frame-rate video camera. For each small video clip we select the central frame as the sharp image and use the frame average as the corresponding blurred image. Finally, to ensure that the averaging process is a sufficient approximation to real blurry images we estimate optical flow and select frames with pixel displacements smaller than a pixel. We demonstrate state of the art performance on datasets with both synthetic and real images.

1 Introduction

This work is concerned with the removal of blur in real images. We consider the challenging case where objects move in an arbitrary way with respect to the camera, and might be occluded and/or come into view. Due to the complexity of this task, prior work has looked at specific cases, where blur is the same everywhere (the shift-invariant case), see *e.g.*, [35, 26], or follows given models [20, 34] and scenarios [15, 28, 38]. Other methods address the modeling complexity by exploiting multiple frames, as in, for example, [16]. Our objective, however, is to produce high-quality results as in [16] by using just a single frame (see Fig. 1). To achieve this goal we use a data-driven approach, where a convolutional neural network is trained on a large number of blurred-sharp image pairs. This approach entails addressing two main challenges: first, the design of a realistic dataset of blurred-sharp image pairs and second, the design of a suitable neural network that can learn from such dataset. We overcome the first challenge by using a commercial high frame-rate video camera (a GoPro Hero5 Black). Due to the high frame-rate, single frames in a video are sharp and motion between frames is small. Then, we use the central frame as the sharp image and the average of all the frames in a video clip as the corresponding blurry image. To avoid averaging frames with too



(a) (b) (c) (d)

Fig. 1: (a) Blurry video frame. (b) Result of [34] on the *single* frame (a). (c) Result of the proposed method on the *single* frame (a). (d) Result of the multi-frame method [16].

much motion, which would correspond to unrealistic motion blurs, we compute the optical flow between subsequent frames and use a simple thresholding strategy to discard frames with large displacements (more than 1 pixel). As we show in the Experiments section, a dataset built according to this procedure allows training a neural network and generalizes to images from other camera models and scenes. To address the second challenge, we build a neural network that replicates (scale-space) pyramid schemes used in classical deblurring methods. The pyramid exploits two main ideas: one is that it is easy to remove a small amount of blur, and the second is that downsampling can be used to quickly reduce the blur amount in a blurry image (within some approximation). The combination of these two contributions leads to a method achieving state of the art performance on the single image space-varying motion blur case.

1.1 Related work

Camera Motion. With the success of the variational Bayesian approach of Fergus et al. [9], a large number of blind deconvolution algorithms have been developed for motion deblurring [2, 5, 25, 35, 29, 26, 44, 41]. Although blind deconvolution algorithms consider blur to be uniform across the image, some of the methods are able to handle small variations due to camera shake [23]. Techniques based on blind deconvolution have been adapted to address blur variations due to camera rotations by defining the blur kernel on a higher dimensional space [11, 12, 38]. Another approach to handle camera shake induced space-varying blur is through region-wise blur kernel estimation [13, 18]. In 3D scenes, motion blur at a pixel is also related to its corresponding depth. To address this dependency, Hu *et al.* and Xu and Jia [15, 42] first estimate a depth map and then solve for the motion blur and the sharp image. In [45], motion blur due to forward or backward camera motion has been explicitly addressed. Notice that blur due to moving objects (see below) cannot be represented by the above camera motion models.
Dynamic Scenes. This category of blur is the most general one and includes motion blur due to camera or object motion. Some prior work [24, 6] addresses this problem by assuming that the blurred image is composed of different regions within which blur is

uniform. Techniques based on alpha matting have been applied to restore scenes with two layers [7, 37]. Although these methods can handle moving objects, they require user interaction and cannot be used in general scenarios where blur varies due to camera motion and scene depth. The scheme of Kim *et al.* [19] incorporates alternating minimization to estimate blur kernels, latent image, and motion segments. Even with a general camera shake model for blurring, the algorithm fails in certain scenarios such as forward motion or depth variations [20]. In [20] Kim and Lee, propose a segmentation-free approach but assume a uniform motion model. The authors propose to simultaneously estimate motion flow and the latent image using a robust total variation (TV-L1) prior. Through a variational-Bayesian formulation, Schelten and Roth [30] recover both defocus as well as object motion blur kernels. Pan *et al.* [27] propose an efficient algorithm to jointly estimate object segmentation and camera motion by incorporating soft segmentation, but require user input. [4, 10, 33] address the problem of segmenting an image into different regions according to blur. Recent works that use multiple frames are able to handle space-varying blur quite well [16, 39].

Deep Learning Methods. The methods in [32, 43] address non-blind deconvolution wherein the sharp image is predicted using the blur estimated from other techniques. In [31], Schuler *et al.* develop an end-to-end system that learns to perform blind deconvolution. Their system consists of modules to extract features, estimate the blur and to perform deblurring. However, the performance of this approach degrades for large blurs. The network of Chakrabarti [3] learns the complex Fourier coefficients of a deconvolution filter for an input patch of the blurry image. Hradiš *et al.* [14] predict clean and sharp images from text documents that are corrupted by motion blur, defocus and noise through a convolutional network without an explicit blur estimation. This approach has been extended to license plates in [36]. [40] proposes to learn a multi-scale cascade of shrinkage fields model. This model however does not seem to generalize to natural images. Sun *et al.* [34] propose to address non-uniform motion blur represented in terms of motion vectors.

Our approach is based on deep learning and on a single input image. However, we directly output the sharp image, rather than the blur, do not require user input and work directly on real natural images in the dynamic scene case. Moreover, none of the above deep learning methods builds a dataset from a high frame-rate video camera. Finally, our proposed scheme achieves state of the art performance in the dynamic scene case.

2 Blurry Images in the Wild

One of the key ingredients in our method is to train our network with an, as much as possible, realistic dataset, so that it can generalize well on new data. As mentioned before, we use a high resolution high frame-rate video camera. We build blurred images by averaging a set of frames. Similar averaging of frames has been done in previous work to obtain data for evaluation [21, 1], but not to build a training set. [21] used averaging to simulate blurry videos, and [1] used averaging to synthesize blurry images, coded exposure images and motion invariant photographs.

We use a handheld GoPro Hero5 Black camera, which captures 240 frames per second with a resolution of 1280×720 pixels. Our videos have been all shot outdoors.

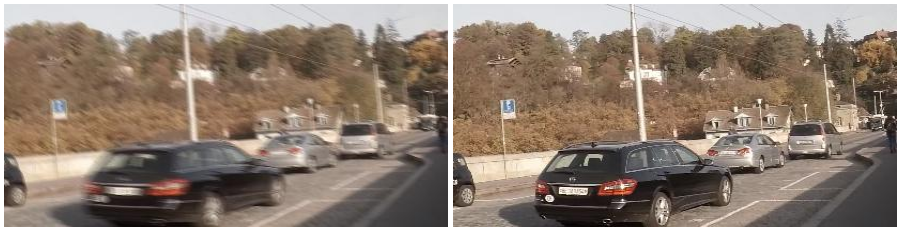


Fig. 2: A sample image pair from the WILD training set. Left: averaged image (the blurry image). Right: central frame (the sharp image).

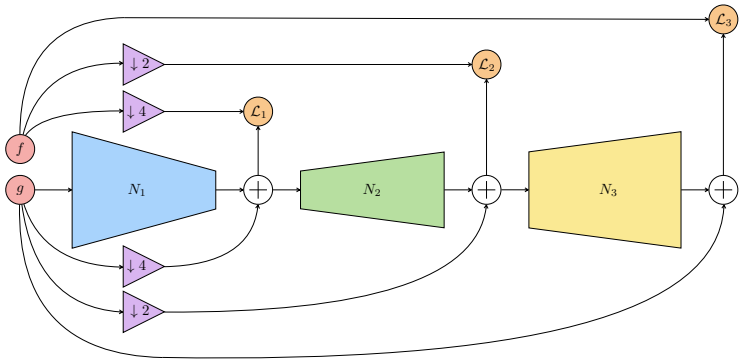


Fig. 3: The DeblurNet architecture. The multiscale scheme allows the network to handle large blurs. Skip connections (bottom links) facilitate the generation of details.

Firstly, we downsample all the frames in the videos by a factor of 3 in order to reduce the magnitude of relative motion across frames. Then, we select the number N_e of averaged frames by randomly picking an odd number between 7 and 23. Out of the N_e frames, the central frame is considered to be the sharp image. We assume that motion is smooth and, therefore, to avoid artifacts in the averaging process we consider only frames where optical flow is no more than 1 pixel. We evaluate optical flow using the recent FlowNet algorithm [8] and then apply a simple thresholding technique on the magnitude of the estimated flow. Fig. 2 shows an example of the sharp and blurred image pair in our training dataset. In this scene, we find both the camera and objects to be moving. We also evaluate when the optical flow estimate is reliable by computing the frame matching error (L^2 norm on the grayscale domain). We found that no frames were discarded in this processing stage (after the previous selection step). We split our *WILD* dataset into training and test sets.

3 The Multiscale Convolutional Neural Network

In Fig. 3 we show our proposed convolutional neural network (CNN) architecture. The network is designed in a pyramid or multi-scale fashion. Inspired by the multi-

	N_1							N_2					N_3				
Type	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	deconv	conv	conv	conv	conv	deconv
OutCh	96	256	384	384	256	256	3	256	256	256	256	3	256	256	256	256	3
Kernel	11	7	7	7	3	3	3	5	5	5	5	5	5	5	5	5	5
Stride	$\downarrow 2$	1	1	$\downarrow 2$	1	1	1	1	1	1	1	$\uparrow 2$	1	1	1	1	$\uparrow 2$

Table 1: The DeblurNet architecture. Batch normalization and ReLU layers inserted after every convolutional layer (except for the last layer of N_1) are not shown for simplicity. Downsampling ($\downarrow$) is achieved by using a stride greater than 1 in convolutional layers. A stride greater than 1 in deconvolutional ($\uparrow$) layers performs upsampling.

scale processing of blind deconvolution algorithms [26, 31], we introduce three subgraphs N_1, N_2, and N_3 in our network, where each subgraph includes several convolution/deconvolution (fractional stride convolution) layers. The task of each subgraph is to minimize the reconstruction error at a particular scale. There are two main differences with respect to conventional CNNs, which play a significant role in generating sharp images without artifacts. Firstly, the network includes a skip connection at the end of each subgraph. The idea behind this technique is to reduce the difficulty of the reconstruction task in the network by using the information already present in the blurry image. Each subgraph needs to only generate a *residual image*, which is then added to the input blurry image (after downsampling, if needed). We observe experimentally that the skip connection technique helps the network in generating more texture details. Secondly, because the extent of blur decreases with downsampling [26], the multi-scale formulation allows the network to deal with small amounts of blur in each subgraph. In particular, the task for the first subgraph N_1 is to generate a deblurred image residual at 1/4 of the original scale. The task for the subgraph N_2 is to use the output of N_1 added to the downsampled input and generate a sharp image at 1/2 of the original resolution. Finally, the task for the subgraph N_3 is to generate a sharp output at the original resolution by starting from the output of N_2 added to the input scaled by $1/2$. We call this architecture the *DeblurNet* and give a detailed description in Tab. 1.

Training. We minimize the reconstruction error of all the scales simultaneously. The loss function $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$ is defined through the following 3 losses

$$\begin{aligned}
\mathcal{L}_1 &= \textstyle\sum_{(g,f)\in\mathscr{D}} \left| N_1(g) + D_{\frac{1}{4}}(g) - D_{\frac{1}{4}}(f) \right|^2 \\
\mathcal{L}_2 &= \textstyle\sum_{(g,f)\in\mathscr{D}} \left| N_2\left(N_1(g) + D_{\frac{1}{4}}(g)\right) + D_{\frac{1}{2}}(g) - D_{\frac{1}{2}}(f) \right|^2 \\
\mathcal{L}_3 &= \textstyle\sum_{(g,f)\in\mathscr{D}} \left| N_3\left(N_2\left(N_1(g) + D_{\frac{1}{4}}(g)\right) + D_{\frac{1}{2}}(g)\right) + g - f \right|^2
\end{aligned} \tag{1}$$

where $\mathscr{D}$ is the training set, g denotes a blurry image, f denotes a sharp image, $D_{\frac{1}{k}}(x)$ denotes the downsampling operation of the image x by factor of k, and N_i indicates the i-th subgraph in the DeblurNet, which reconstructs the image at the i-th scale.

Implementation Details. We used Adam [22] for optimization with momentum parameters as $\beta_1 = 0.9$, $\beta_2 = 0.999$, and an initial learning rate of 0.001. We decrease the learning rate by $.75$ every 10^4 iterations. We used 2 Titan X for training with a batch size of 10. The network needs 5 days to converge using batch normalization [17].

4 Experiments

We tested DeblurNet on three different types of data: a) the WILD test set (GoPro Hero5 Black), b) real blurry images (Canon EOS 5D Mark II), and c) data from prior work.

Synthetic vs pseudo-real training. To verify the impact of using our proposed averaging to approximate space-varying blur, we trained another network with the same architecture as in Fig. 3. However, we used blurry-sharp image pairs, where the blurry image is obtained synthetically via a shift-invariant convolutional model. As in [3], we prepared a set of 10^5 different blurs. During training, we randomly pick one of these motion blurs and convolve it with a sharp image (from a mixture of $50K$ sharp frames from our WILD dataset and $100K$ cityscapes images[1]) to generate blurred data. We refer to this trained network as the *DeblurNet*$^{\text{SI}}$, where *SI* stands for shift-invariant blur. A second network is instead trained only on the blurry-sharp image pairs from our WILD dataset (a total of $50K$ image pairs obtained from the selection and averaging process on the GoPro Hero5 Black videos). This network is called *DeblurNet*$^{\text{WILD}}$, where *WILD* stands for the data from the WILD dataset. As will be seen later in the experiments, the *DeblurNet*$^{\text{WILD}}$ network outperforms the *DeblurNet*$^{\text{SI}}$ network despite the smaller training set and the fact that the same sharp frames from the WILD dataset have been used. Therefore, due to space limitations, often we will show only results of the *DeblurNet*$^{\text{WILD}}$ network in the comparisons with other methods.

WILD test set evaluation. The videos in the test set were captured at locations different from those where training data was captured. Also, incidentally, the weather conditions during the capture of the test set were significantly different from those of the training set. We randomly chose 15 images from the test-set and compared the performance of our method against the methods in [41], [34], the space-varying implementation of the method in [44], and *DeblurNet*$^{\text{WILD}}$ trained network. An example image is shown in Fig. 4. As can be observed, blur variation due to either object motion or depth changes is the major cause of artifacts. Our *DeblurNet*$^{\text{WILD}}$ network, however, produces artifact-free sharp images. While the example in Fig. 4 gives only a qualitative evaluation, in Table 2 we report quantitative results.

We measure the performance of all the above methods in terms of Peak Signal-to-Noise Ratio (PSNR) by using the reference sharp image as in standard image deblurring performance evaluations. We can see that the performance of the *DeblurNet*$^{\text{WILD}}$ is better than that of the *DeblurNet*$^{\text{SI}}$. This is not surprising because the shift-invariant training set does not capture factors such as reflections/specularities, the space-varying blur, occlusions and coming into view of objects. Notice that the PSNR values are not comparable to those seen in shift-invariant deconvolution algorithms.

[34]	[41]	[44]	*DeblurNet*$^{\text{SI}}$	*DeblurNet*$^{\text{WILD}}$
25.48	23.61	22.50	25.8	**28.1**

Table 2: Average PSNR on our WILD test set.

Qualitative evaluation. On other available *dynamic scene blur* datasets the ground truth is not available. Therefore, we can only evaluate our proposed network qualita-

[1] www.cityscapes-dataset.com

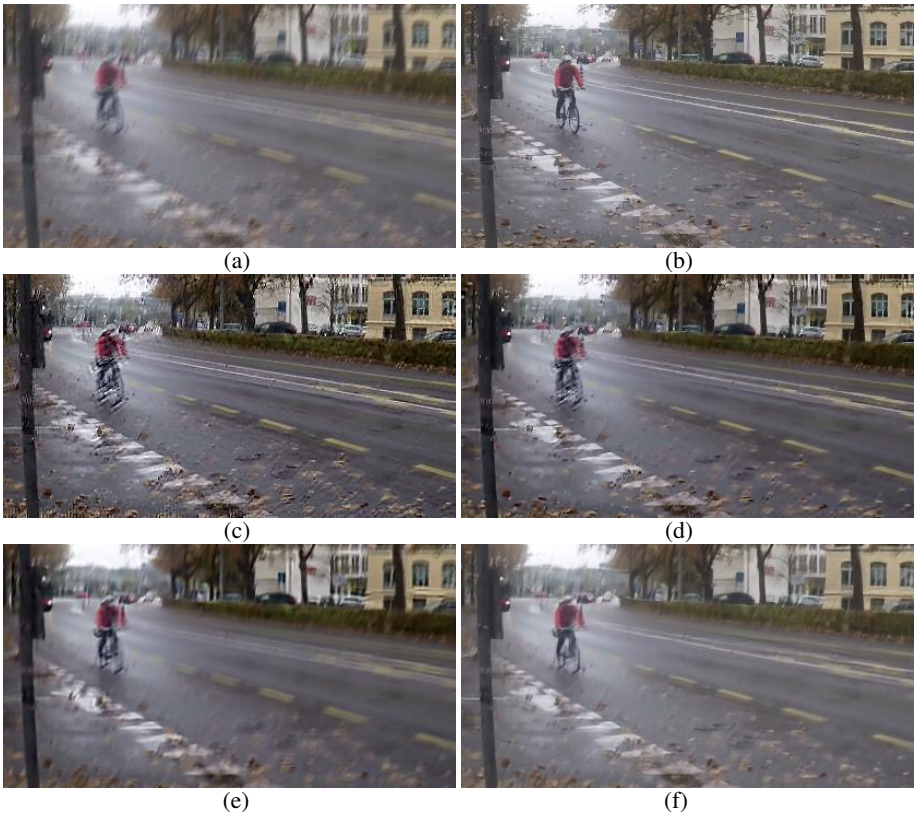


Fig. 4: An example from the WILD test set. (a) blurry image, (b) sharp image (ground truth), (c) Xu and Jia [41], (d) Xu *et al.* [44], (e) Sun *et al.* [34], (f) *DeblurNet*$^{\text{WILD}}$.

tively. We consider 2 available datasets and images obtained from a Canon EOS 5D Mark II camera. While Figs. 5 and 7 show data from [34] and [20] respectively, Fig. 6 shows images from the Canon camera. In Fig. 6, we compare the methods of [41], [34] and [44] to both our *DeblurNet*$^{\text{SI}}$ and *DeblurNet*$^{\text{WILD}}$ networks. In all datasets, we observe that our method is able to return sharper images with fine details. Furthermore, we observe that in Fig. 6 the *DeblurNet*$^{\text{WILD}}$ network produces better results than the *DeblurNet*$^{\text{SI}}$ network, which confirms once more our expectations.

Shift-invariant blur evaluation. We provide a brief analysis on the differences between dynamic scene deblurring and shift-invariant motion deblurring. We use an example from the standard dataset of [23], where blur is due to camera shake (see Fig. 8). In the case of a shift-invariant blur, there are infinite $\{$blur, sharp image$\}$ pairs that yield the same blurry image when convolved. More precisely, an unknown 2D translation (shift) in a sharp image f can be compensated by an opposite 2D translation in the blur kernel k, that is, $\forall \Delta$, $g(x) = \int f(y + \Delta)k(x - y - \Delta)dy$. Because of such ambiguity, current evaluations compute the PSNR for all possible 2D shifts of f and pick the

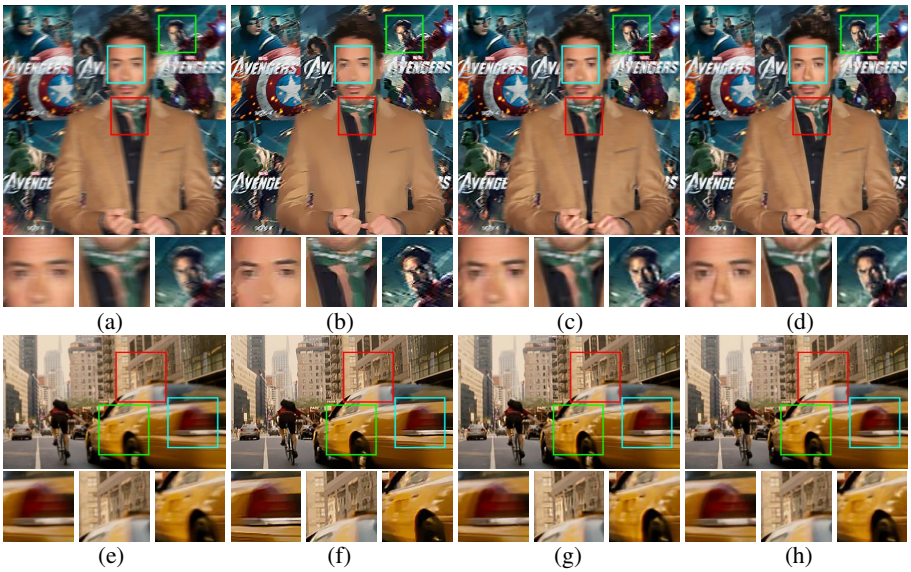


Fig. 5: Test set from [20]. (a,e) Blurry image; (b,f) Kim and Lee [20]; (c,g) Sun *et al.* [34]; (d,h) *DeblurNet*$^{\text{WILD}}$.

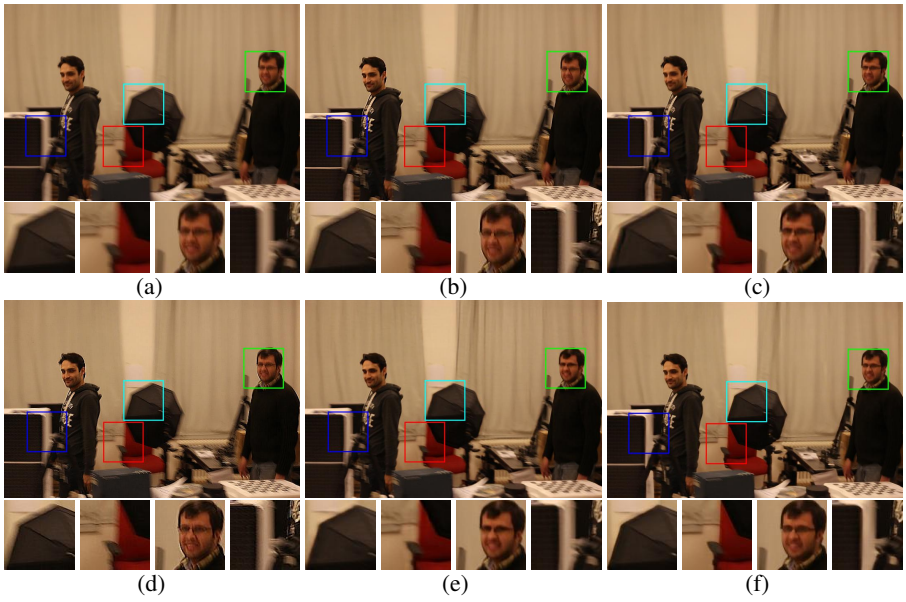
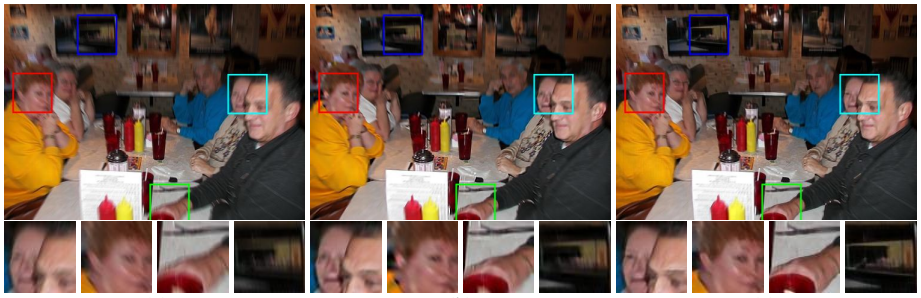


Fig. 6: Test set from the Canon camera. (a) Blurry image; (b) Xu *et al.* [44]; (c) Sun *et al.* [34]; (d) Xu and Jia [41]; (e) *DeblurNet*$^{\text{SI}}$; (f) *DeblurNet*$^{\text{WILD}}$.



(a) (b) (c)

Fig. 7: Test dataset from [34]. (a) Blurry image, (b) Sun *et al.* [34], (c) *DeblurNet*$^{\text{WILD}}$.

(a) (b) (c) (d)

Fig. 8: Kohler dataset [23] (image 1, blur 4). (a) our result. (b) ground truth. (c,d) Zoomed-in patches. Local ambiguous shifts are marked with white arrows.

highest PSNR. The analogous search is done for camera shake [23]. However, with a dynamic scene we have ambiguous shifts at every pixel (see Fig. 8) and such search is unfeasible (the image deformation is undefined). Therefore, all methods for dynamic scene blur would be at a disadvantage with the current shift-invariant blur evaluation methods, although their results might look qualitatively good.

Analysis. Our network generates a residual image that when added to the blurry input yields the sharp image. Therefore, we expect the magnitude of the residual to be large for very blurry images, as more changes will be required. To validate this hypothesis we perform both quantitative and qualitative experiments. We take 700 images from another WILD test set (different from the 15 images used in the previous quantitative evaluation), provide them as input to the *DeblurNet*$^{\text{WILD}}$ network, and calculate the L^1 norm of the network residuals (the output of the last layer of N_3). In Fig. 10 we show two images, one with the highest and one with the lowest L^1 norm. We see that the residuals with the highest norms correspond to highly blurred images, and vice versa for the low norm residuals. We also show quantitatively that there is a clear correlation between the amount of blur and the residual L^1 norm. As mentioned earlier on, our WILD dataset also computes an estimate of the blurs by integrating the optical flow. We use this blur estimate to calculate the average blur size across the blurry image. This gives us an approximation of the overall amount of blur in an image. In Fig. 9 we show the plot of the L^1 norm of the residual versus the average estimated blur size for all 700 images. The residual magnitudes and blur sizes are normalized so that mean and standard deviation are 0 and 1 respectively.

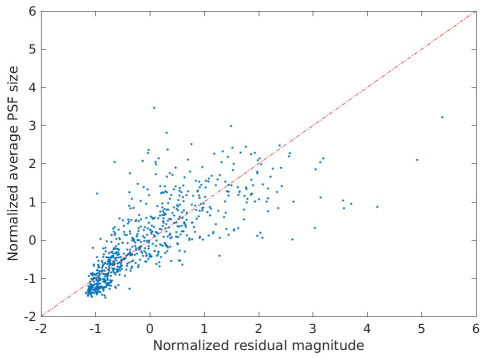


Fig. 9: Normalized average blur size versus normalized residual magnitude plot. Notice the high level of correlation between the blur size and the residual magnitude.

Fig. 10: The images with highest (first row) and lowest (second row) residual norm in the output layer. The image in the first column is the input, the second column shows the estimated residual (the network output), the third column is the deblurred image (first column + second column), and finally the forth column is the ground truth.

5 Conclusions

We proposed DeblurNet, a novel CNN architecture that regresses a sharp image given a blurred one. DeblurNet is able to restore blurry images under challenging conditions, such as occlusions, motion parallax and camera rotations. The network consists of a chain of 3 subgraphs, which implement a multiscale strategy to break down the complexity of the deblurring task. Moreover, each subgraph outputs only a residual image that yields the sharp image when added to the input image. This allows the subgraph to focus on small details as confirmed experimentally. An important part of our solution is the design of a sufficiently realistic dataset. We find that simple frame averaging combined with a very high frame-rate video camera produces reasonable blurred-sharp image pairs for the training of our DeblurNet network. Indeed, both quantitative and qualitative results show state of the art performance when compared to prior dynamic scene deblurring work. We observe that our network does not generate artifacts, but may leave extreme blurs untouched.

Acknowledgements. Paolo Favaro acknowledges support from the Swiss National Science Foundation on project 200021_153324.

References

1. Agrawal, A., Raskar, R.: Optimal single image capture for motion deblurring. In: CVPR (2009)
2. Babacan, S.D., Molina, R., Do, M.N., Katsaggelos, A.K.: Bayesian blind deconvolution with general sparse image priors. In: ECCV (2012)
3. Chakrabarti, A.: A neural approach to blind motion deblurring. In: ECCV (2016)
4. Chakrabarti, A., Zickler, T., Freeman, W.T.: Analyzing spatially-varying blur. In: CVPR (2010)
5. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graph. 28(5), 1–8 (2009)
6. Couzinie-Devy, F., Sun, J., Alahari, K., Ponce, J.: Learning to estimate and remove non-uniform image blur. In: CVPR (2013)
7. Dai, S., Wu, Y.: Removing partial blur in a single image. In: CVPR (2009)
8. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: CVPR (2015)
9. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graph. 25(3), 787–794 (2006)
10. Gast, J., Sellent, A., Roth, S.: Parametric object motion from blur. arXiv preprint arXiv:1604.05933 (2016)
11. Gupta, A., Joshi, N., Zitnick, L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: ECCV (2010)
12. Hirsch, M., Sra, S., Schölkopf, B., Harmeling, S.: Efficient filter flow for space-variant multiframe blind deconvolution. In: CVPR (2010)
13. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: ICCV (2011)
14. Hradiš, M., Kotera, J., Zemcík, P., Šroubek, F.: Convolutional neural networks for direct text deblurring. In: BMVC (2015)
15. Hu, Z., Xu, L., Yang, M.H.: Joint depth estimation and camera shake removal from single blurry image. In: CVPR (2014)
16. Hyun Kim, T., Mu Lee, K.: Generalized video deblurring for dynamic scenes. In: CVPR (2015)
17. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
18. Ji, H., Wang, K.: A two-stage approach to blind spatially-varying motion deblurring. In: CVPR (2012)
19. Kim, T.H., Ahn, B., Lee, K.M.: Dynamic scene deblurring. In: ICCV (2013)
20. Kim, T.H., Lee, K.M.: Segmentation-free dynamic scene deblurring. In: CVPR (2014)
21. Kim, T.H., Nah, S., Lee, K.M.: Dynamic scene deblurring using a locally adaptive linear blur model. arXiv preprint arXiv:1603.04265 (2016)
22. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Köhler, R., Hirsch, M., Mohler, B.J., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: ECCV (2012)
24. Levin, A.: Blind motion deblurring using image statistics. In: NIPS (2006)
25. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Efficient marginal likelihood optimization in blind deconvolution. In: CVPR (2011)
26. Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: ECCV (2014)

27. Pan, J., Hu, Z., Su, Z., Lee, H.Y., Yang, M.H.: Soft-segmentation guided object motion deblurring. In: CVPR (2016)
28. Paramanand, C., Rajagopalan, A.N.: Non-uniform motion deblurring for bilayer scenes. In: CVPR (2013)
29. Perrone, D., Favaro, P.: Total variation blind deconvolution: The devil is in the details. In: CVPR (2014)
30. Schelten, K., Roth, S.: Localized image blur removal through non-parametric kernel estimation. In: ICPR (2014)
31. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(7), 1439–1451 (2016)
32. Schuler, C.J., Christopher Burger, H., Harmeling, S., Scholkopf, B.: A machine learning approach for non-blind image deconvolution. In: CVPR (2013)
33. Shi, J., Xu, L., Jia, J.: Discriminative blur detection features. In: CVPR (2014)
34. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: CVPR (2015)
35. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: ICCP (2013)
36. Svoboda, P., Hradiš, M., Maršík, L., Zemcík, P.: Cnn for license plate motion deblurring. In: ICIP (2016)
37. Tai, Y.W., Kong, N., Lin, S., Shin, S.Y.: Coded exposure imaging for projective motion deblurring. In: CVPR (2010)
38. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: CVPR (2010)
39. Wieschollek, P., Schölkopf, B., Lensch, H.P.A., Hirsch, M.: End-to-end learning for image burst deblurring. In: ACCV (2016)
40. Xiao, L., Wang, J., Heidrich, W., Hirsch, M.: Learning high-order filters for efficient blind deconvolution of document photographs. In: ECCV (2016)
41. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: ECCV (2010)
42. Xu, L., Jia, J.: Depth-aware motion deblurring. In: ICCP (2012)
43. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: NIPS (2014)
44. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: CVPR (2013)
45. Zheng, S., Xu, L., Jia, J.: Forward motion deblurring. In: CVPR (2013)