# Burst Image Deblurring Using Permutation Invariant Convolutional Neural Networks

Miika Aittala and Frédo Durand

Massachusetts Institute of Technology, Cambridge MA 02139, USA
miika@csail.mit.edu, fredo@mit.edu

**Abstract.** We propose a neural approach for fusing an arbitrary-length burst of photographs suffering from severe camera shake and noise into a sharp and noise-free image. Our novel convolutional architecture has a simultaneous view of all frames in the burst, and by construction treats them in an order-independent manner. This enables it to effectively detect and leverage subtle cues scattered across different frames, while ensuring that each frame gets a full and equal consideration regardless of its position in the sequence. We train the network with richly varied synthetic data consisting of camera shake, realistic noise, and other common imaging defects. The method demonstrates consistent state of the art burst image restoration performance for highly degraded sequences of real-world images, and extracts accurate detail that is not discernible from any of the individual frames in isolation.

**Keywords:** burst imaging · image processing · deblurring · denoising · convolutional neural networks

## 1 Introduction

Motion blur and noise remain a significant problem in photography despite advances in light efficiency of digital imaging devices. Mobile phone cameras are particularly suspect to handshake and noise due to the small optics and the typical unsupported free-hand shooting position. Shortcomings of optical systems can be in part ameliorated by computational procedures such as denoising and sharpening. One line of work that has recently had significant impact relies on *burst imaging*. A notable example is the imaging pipeline supplied in Android mobile phones: transparently to the user, the camera shoots a sequence of low-quality frames and fuses them computationally into a higher-quality photograph than could be achieved with a conventional exposure in same time [12].

We address the problem of *burst deblurring*, where one is presented with a set of images depicting the same target, each suffering from a different realization of

To appear in Springer LNCS series (ECCV 2018 proceedings). This is an author-prepared preprint. Supplemental material is available at http://people.csail.mit.edu/miika/eccv18_deblur/

camera shake. While each frame might be hopelessly blurred in isolation, they still retain pieces of partial information about the underlying sharp image. The aim is to recover it by fusing whatever information is available.

Convolutional neural networks (CNN's) have led to breakthroughs in a wide range of image processing tasks, and have also been applied to burst deblurring [34,33]. Observing that bursts can have arbitrarily varying lengths, the recent work of Wieschollek et al. [33] maintains an estimate of the sharp image, and updates it in a recurrent manner by feeding in the frames one at a time. While this is shown to produce good results, it is well known that recurrent architectures struggle with learning to fuse information they receive over a number of steps – even a task as simple as summing together a set of numbers can be difficult [36]. Indeed, our evaluation shows that the architecture of Wieschollek et al. [33] fails to e.g. fully use a lucky sharp image present in a burst (see Figure 7). This suggests that it generally does not make full use of the information available.

The problem, we argue, is that a recurrent architecture puts the different frames into a highly asymmetric position. The first and the most recently seen frames can have a disproportionate influence on the solution, and complementary cues about individual image details are difficult to combine if they appear multiple frames apart.

We propose a fundamentally different architecture, which considers all of the frames simultaneously as an *unordered set* of arbitrary size. The key idea is to enforce *permutation invariance* by construction: when the ordering of the frames cannot affect the output, no frame is in a special position in relation to others, and consequently each one receives the same consideration. Any piece of useful information can directly influence the solution, and subtle cues scattered around in the burst can be combined effectively. The approach is similar in spirit to classical maximum likelihood or Bayesian inference, where contributions from each observation are symmetrically accumulated onto a likelihood function, from which the desired estimate is then derived.

We achieve this by extending recent ideas on permutation invariance in neural networks [36,24] to a convolutional image translation context. Our proposed network is a U-Net-inspired [25] CNN architecture that maps an unordered set of images into a single output image in a perfectly permutation-invariant manner, and facilitates repeated back-and-forth exchanges of feature information between the frames during the network evaluation. Besides deblurring, we believe that this general-purpose architecture has potential applications to a variety of problems involving loosely structured sets of image-valued observations.

We train our network with synthetically degraded bursts consisting of a range of severe image defects beyond just blur. The presence of noise changes the character of the deblurring problem, and in practice many deblurring algorithms struggle with the high noise levels in full-resolution low-light photographs, and images from low-end cameras. Of course, these are exactly the scenarios where deblurring would be most needed. Our training data simulates the noise characteristics of real-world cameras, and also considers some often overlooked details such as unknown gamma correction, and high dynamic range effects.

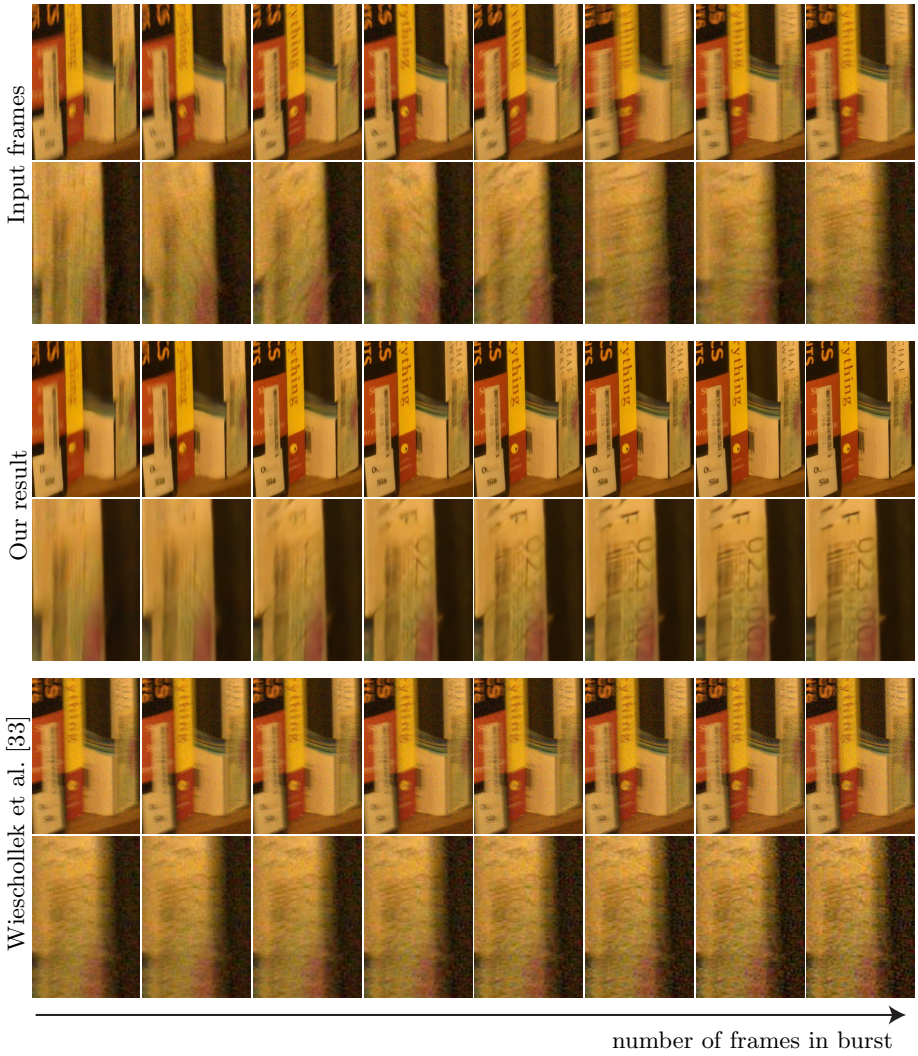



**Fig. 1.** Given a burst of full-resolution mobile phone camera images suffering from significant camera shake and noise (top rows), our method recovers a sharp estimate of the underlying image (middle rows). The horizontal sequence shows our solution for a growing number of input frames; the rightmost result uses all eight images. The bottom row shows the same progression for the state of the art neural burst deblurring method of Wieschollek et al. [33] (computed using their software implementation). Note how our method has resolved details that are difficult or impossible to reliably discern by eye in any of the inputs: for example the numbers 023-002 at the right edge (shown in blow-up) appear in the solution somewhere around the fourth frame, and become gradually sharper as images are added. These are the actual numbers on the subject (see the supplemental material for a verification photo). Note also the substantial reduction in noise. Images in this paper are best viewed digitally.

Figure 1 demonstrates the effectiveness of our approach compared to the state of the art recurrent architecture of Wieschollek et al. [33] on a challenging real-world burst involving significant image degradations: our method successfully recovers image content that appears to be all but lost in the individual frames of the burst, and markedly improves the overall image quality.

## 2 Related work

### 2.1 Image restoration

**Deblurring** Restoring sharp images from blurred observations is a long-standing research topic. Wang and Tao [32] present a recent survey of approaches to this problem. Deconvolution algorithms seek to stably invert the linear convolution operation when the blur kernel is known. *Blind* deconvolution concerns the more challenging case when the kernel is unknown [10,20,2].

Various methods use neural networks to estimate blur kernels from images and apply classical non-blind deconvolution algorithms to perform the actual deblurring, either as a separate step or as an integrated part of the network [31,28,4,35]. Other approaches sidestep the classical deconvolution, and train a CNN to output a sharp image directly. Nah et al. [22] and Noroozi et al. [23] deblur single images using multi-scale end-to-end convolutional architectures.

Nah et al. [22] and Kupyn et al. [19] use a discriminator-based loss, which encourages the network to produce realistic-looking content in the deblurred image. The downside is that the network may in principle need to invent fictional detail in order to achieve an appearance of realism. We view this direction as largely orthogonal to ours, and focus on extracting the maximum amount of real information from the input burst.

**Multi-frame methods** A variety of deblurring methods consider combining information from multiple blurred frames [3,29,38,37,39]. Delbracio et al. [7] showed that for static scenes, the typical difficulties with multi-frame blind deconvolution can be sidestepped by combining the well-preserved parts of the power spectra of each frame in the Fourier domain. Wieschollek et al. [34] extend this by determining the deconvolution filters and the spectral averaging weights using a neural network. Recently Wieschollek et al. [33] proposed a neural method for directly predicting a sharp image from an arbitrary-length burst of blurred images, by using a recurrent neural architecture that updates the estimate when fed with new frames. Our method targets the same problem with a fundamentally different network architecture that instead considers all images simultaneously as a *set* of arbitrary size.

Some methods also aim to remove local blur caused by movement of individual objects in videos [30,33,15,5]. Our method focuses on blur caused by camera shake, where the correlations between frames are weak and the frame ordering carries a minor significance. We nonetheless demonstrate that in practice our approach is applicable to flow-aligned general motion data of Su et al. [30].

Multi-frame burst ideas have recently also been applied to denoising [12,11,21]. While our main concern is deblurring, we train our model with heavily noisy images to promote robustness against real-world imaging defects. Consequently, our method also learns to denoise the bursts to a significant degree.

### 2.2 Permutation invariance

A wide range of inference problems concern unordered sets of input data. For example, point clouds in Euclidian space have no natural ordering, and consequently any global properties we compute from them should not depend on what order the points are provided in. The same holds for inferences made from i.i.d. (or more generally, exchangeable) realizations of random variables, as is the case for example in maximum likelihood and Bayesian estimation.

For neural networks, switching the places of a pair of inputs generally changes the output, and for a good reason: the particular arrangement of the pixels in an image is strongly indicative of the subject depicted, and the meaning of a sentence depends on the order of the words. This is, however, problematic with set-valued data, because one cannot opt out of assigning an ordering. This is counterproductive, as the network will attempt to attribute some meaning to the order. A common argument is that in practice the network should learn that the input order is irrelevant, but this claim is both theoretically unsatisfying and empirically dubious. It is possible that permutation invariance is not easily learnable, and a significant amount of network capacity must be allocated towards achieving an approximation of it.

A variety of recent works have recognized this shortcoming and proposed architectures that handle unordered inputs in a principled way. Zaheer et al. [36] analyze the general characteristics of set-valued functions, and present a framework for constructing permutation invariant neural networks by the use of symmetric pooling layers. Qi et al. [24] propose a similar pooling architecture on point cloud data. Edwards and Storkey [8] use symmetric pooling for the purpose of learning to extract meaningful statistics out of datasets. Herzig et al. [13] apply similar ideas to achieve permutation invariance in structured scene graphs describing hierarchical relations of objects in an image. Korshunova et al. [18] address learning to generalize from a set of observations via a provably permutation invariant recurrent architecture. We discuss these ideas in detail in Section 3.1 and extend them to image translation CNN's.

## 3 Method

Our method consists of a convolutional neural network, which outputs a restored image when fed with a set of blurry and noisy images that have been approximately pre-aligned using homographies. We describe the network architecture in Section 3.1, and the synthetic data generation pipeline we use for training it in Section 3.2.

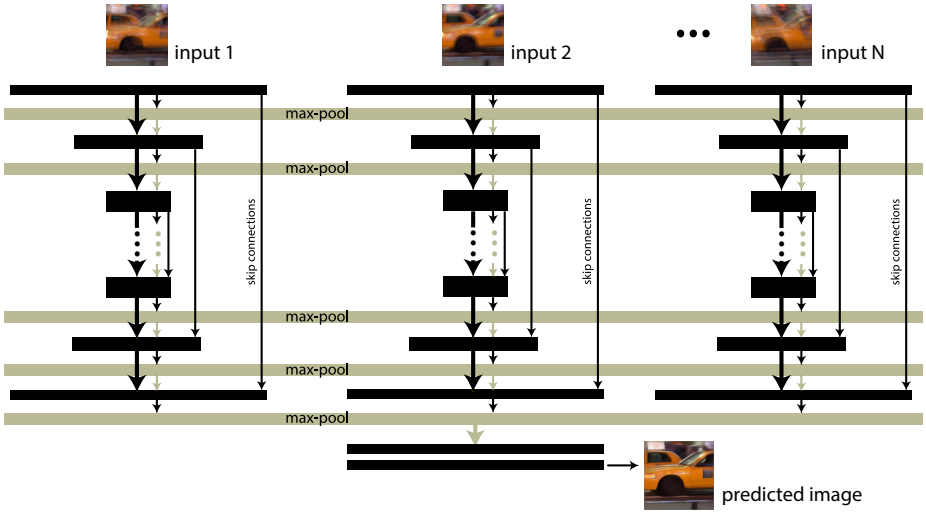



**Fig. 2.** Overview of our network architecture. Each input frame is processed by a copy of the same U-Net [25] with tied weights, but information is repeatedly exchanged between the copies. This is achieved by computing the maximum value of each activation between all the tracks, and concatenating these "global features" back into the per-frame local features. After the encoder-decoder cycle, the tracks are collapsed by a final max-pooling and processed into a joint estimate of the clean image. Observe that permuting the ordering of the inputs cannot change the output, and that their number can be arbitrary. See Figure 3 for a detailed view of the layers.

### 3.1 Network architecture

Zaheer et al. [36] and Qi et al. [24] show that any function that maps an unordered set into a regular vector (or an image) can be approximated by a neural network as follows. The individual members of the set are first processed separately by *identical* neural networks with tied weights, yielding a vector (or an image) of features for each of them. The features are then *pooled* by a symmetric operation, by evaluating either the mean or maximum value of each feature across the members. That is, if the $i$'th output feature for the $k$'th member in the set is denoted as $x_i^k$, then a max-pooling operation returns the features $x_i^{\text{pooled}} = \max_k x_i^k$. The individual members are then forgotten, and the pooled features are processed by further neural network layers in the regular fashion. The key idea is that through end-to-end training, the per-member network will learn to output features for which the pooling is meaningful; intuitively, the pooling acts as a "vote" over the joint consensus on the global features. The remaining layers then extract the desired output from this consensus. Note that the symmetry of the pooling makes this scheme perfectly permutation invariant, and indifferent to the cardinality of the input set.

In our context, this scheme gives the individual frames in the burst a principled mechanism for contributing their local findings about the likely content

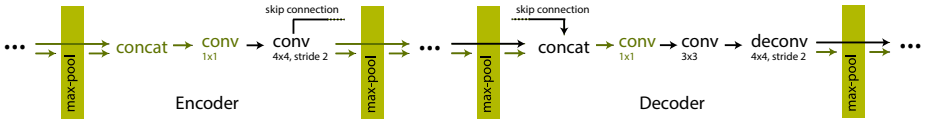



**Fig. 3.** A zoomed-in view of a single encoder downsampling unit (left) and a corresponding decoder upsampling unit (right) for a U-Net in Figure 2, connected by a skip connection. The green-colored nodes indicate the layers we introduce. The max-pool layers transmit information to and from other tracks; notice that without them the architecture reduces to a regular U-Net. We use the exponential linear unit nonlinearity [6] in all layers except the final one.

of the sharp image. We apply it on a U-Net-style architecture [25], which is a proven general-purpose model for transforming images [14]. The U-Net is a hourglass-shaped network consisting of an "encoder" that sequentially reduces the image to a low resolution, and a "decoder" that expands it back into a full image. Skip connections are used between correspondingly sized layers in the encoder and decoder to aid reconstruction of details at different scales.

Our high-level architecture is illustrated in Figure 2: as discussed above, each of the inputs is processed by a tied copy of the same U-Net, and the results are max-pooled towards the end and further processed into an estimate of the sharp image. We additionally introduce intermediate pooling layers followed by concatenation of the pooled "global state" back into the local features. This enables repeated back-and-forth information exchanges between the members of the set in a permutation equivariant manner. This is achieved at a relatively low additional computational cost by fusing the global features into the local features with a $1 \times 1$ convolution after each pooling. See Figure 3 for a more detailed illustration of the layer connections in individual units of the U-Net. The exact details are available in the supplemental appendix and the associated code release. We also experimented with mean-pooling in place of max-pooling, and found the performance of this variant to be similar.

Note that by omitting the final pooling one ends up instead with a method for translating image sets to image sets in a permutation equivariant manner. While we don't make use of this variant, it may have applicability to other problems.

### 3.2 Training data

We train our method with bursts of synthetically degraded crops of photographs from the Imagenet [26] dataset. The degradations are generated on the fly in TensorFlow [1]. Severity and intra-burst variation of the effects is randomized, so as to encourage the network to robustly take advantage of different cues. We also consider noise with inter-pixel correlations, unknown gamma correction, and streaks caused by blurring of overexposed image regions.

We generate training pairs of resolution $160 \times 160$, where the input is a degraded burst, and the target is a corresponding clean image. The length of the burst is randomized between 1 and 8 for each minibatch. Figure 4 shows

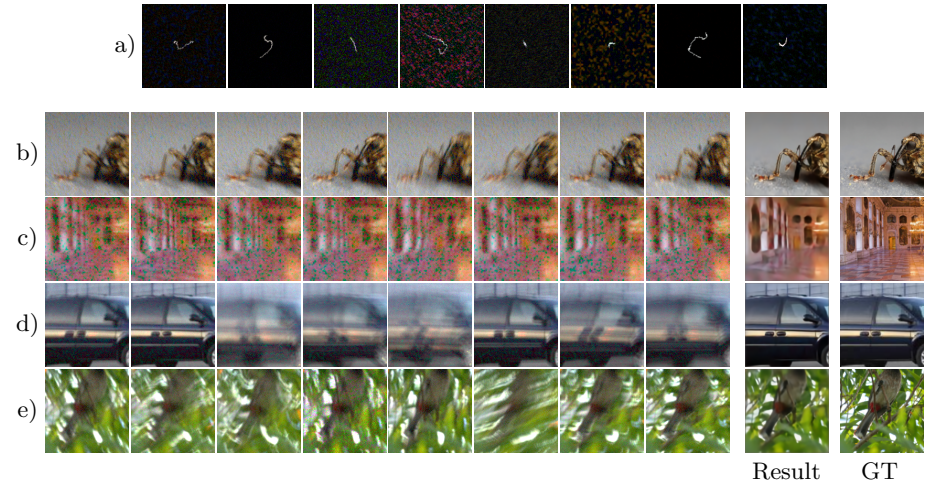



**Fig. 4.** (a): Blur kernels and noises generated by our synthetic training data generation pipeline. Notice that many of the kernels span several dozen pixels. (b)–(e): Synthetically degraded bursts from a held-out validation set, along with our network's prediction and the ground truth target. Notice the varying difficulty of the bursts, and the streaks from our dynamic range expansion scheme on the saturated skylight (e).

examples of individual kernels and noises, as well as full degraded bursts from our pipeline. We give an overview of each component below.

**Kernel generation** We simulate camera shake by convolving the clean photographs with random kernels. Each kernel is generated as a random walk of 128 steps by first drawing 2D acceleration vectors from unit normal distribution, and integrating them into velocities and positions by a pair of (damped) cumulative sums, taking care to choose the initial velocity from the stationary distribution.

We then center each kernel at the origin to avoid random misalignment between the frames and the training target, and standardize them to a unit variance. We apply a random scale to the entire burst's kernels, and then scale individual kernels in the burst randomly. The individual variations are randomized so that some bursts have uniformly sized kernels and others are mixtures of small and large ones. To encourage modest defocus deblurring, we perturb the points with small random offsets. Finally, the random walk positions are accumulated into bitmaps of size $51 \times 51$ by an additive scattering operation, yielding the desired convolution kernels.

**Noise generation** Various factors introduce pixel correlations in imaging noise and give it a characteristic "chunky" appearance: Bayer interpolation, camera software's internal denoising, smearing between pixels as they are alignined, compression, and so on. We simulate these effects with a heuristic noise-generation

pipeline that mimics the visual appearance of typical camera noise. To this end, we feed an i.i.d. Gaussian noise image through a combination of random convolutions, ReLU nonlinearities and random up- and downsamplings, and apply it in random proportions additively and multiplicatively.

**Other imaging effects** We target our method for real-world images that have gone through unknown gamma correction and color processing. Because motion blur occurs in linear space, we linearize the synthetic images prior to blurring, and re-apply the gamma correction afterwards. As we do not know the true gamma value for each Imagenet image, we simply pick a random value between 1.5 and 2.5. This procedure introduces the correct kind of post-blur nonlinearity, and encourages robustness against a variety of unknown nonlinearities the input image may have suffered. At test time, we perform no gamma-related processing.

Visible light sources and bright highlight regions often appear as elongated streaks when blurred (see e.g. Figure 10c). This effect is not reproduced in synthetically blurred low dynamic range images, as their pixel intensities are saturated at a relatively low value. We reintroduce fictional high dynamic range content prior to blurring by adding intensity-boosted image data onto saturated regions from other images in the same minibatch. After blurring, we clip the image values again. The effect is often surprisingly convincing (see Figure 4e).

### 3.3 Technical details

We use the loss function $L(a, b) = \frac{1}{10}||a - b||_1 + ||\nabla a - \nabla b||_1$, where $\nabla$ computes the (unnormalized) horizontal and vertical direction finite differences. This weighting assigns extra importance on reconstructing image edges.

We use weight normalization [27] and the associated data-dependent initialization scheme on all layers to stabilize the training.

The method is implemented in TensorFlow [1]. We train the model using the Adam [16] optimization algorithm with a learning rate of 0.003 and per-iteration decay of 0.999997. We train for 400 000 iterations with minibatch size of 8 split across two NVIDIA GTX 1080 Ti GPU's. This takes roughly 55 hours.

For large images, we apply the network in overlapping sliding windows, with smooth blending across the overlap. The downscaling cycle of the U-Net gives the network a moderately wide receptive field. The method naturally handles inputs for which the kernel slowly varies across the image.

The runtime depends roughly linearly on both the number of input images as well as their pixel count. For a 12-megapixel 8-frame burst, the evaluation takes 1.5 minutes on a single GPU. The model has approximately 40M parameters.

The burst frames are pre-aligned with homographies using dense correspondences with the ECC algorithm [9]. This takes around 18 seconds per 12-megapixel frame. Pixel-perfect alignment does not appear to be critical; a small amount of parallax can be seen in many of our evaluation datasets.

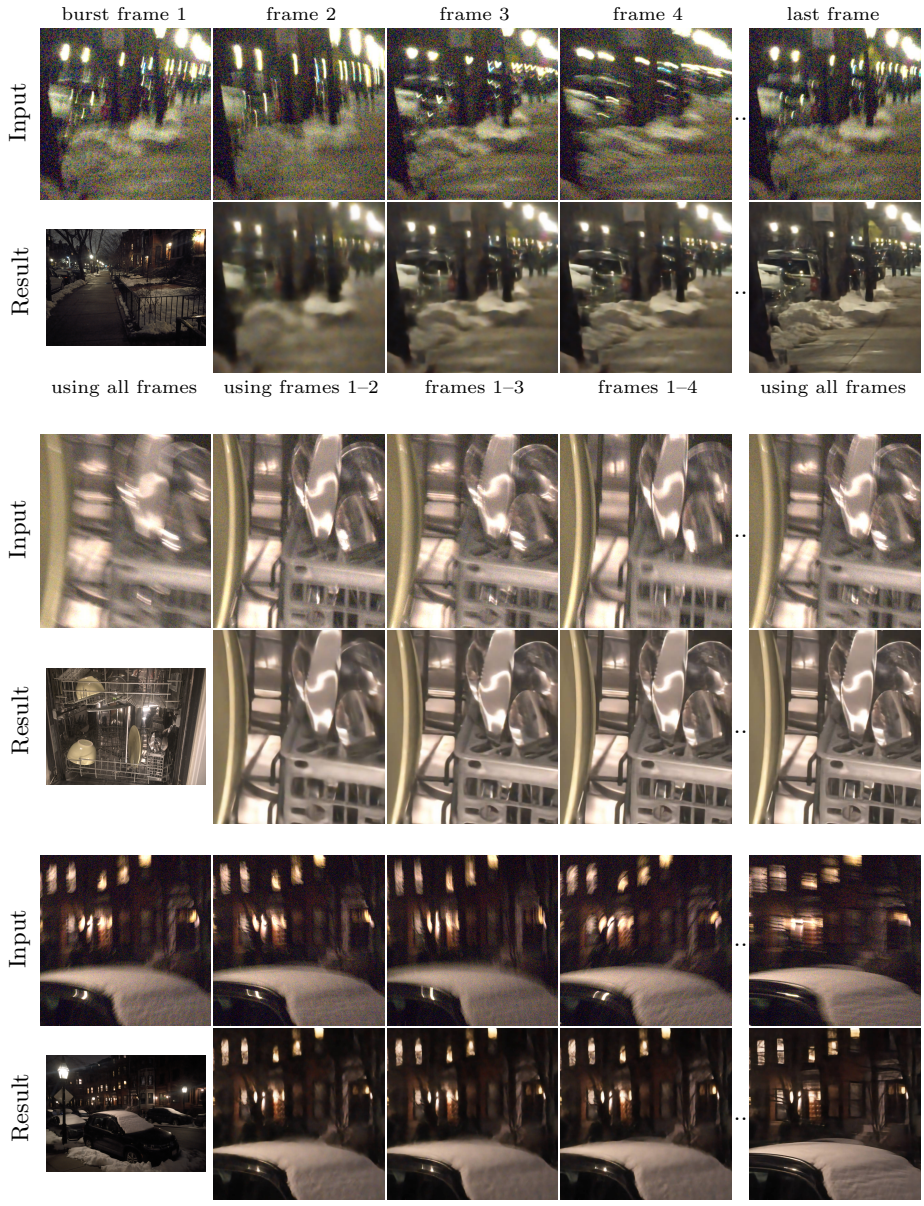



**Fig. 5.** A selection of results for challenging bursts from a mobile phone camera in full resolution. The full lengths of these bursts are 12, 5 and 10 frames, respectively. We show the first four and the last input, as well as the full-resolution output for the entire burst, and crops of intermediate and full burst outputs. Please refer to the supplemental material for full-size images, as well as results from recurrent neural model of Wieschollek et al. [33] on these images.

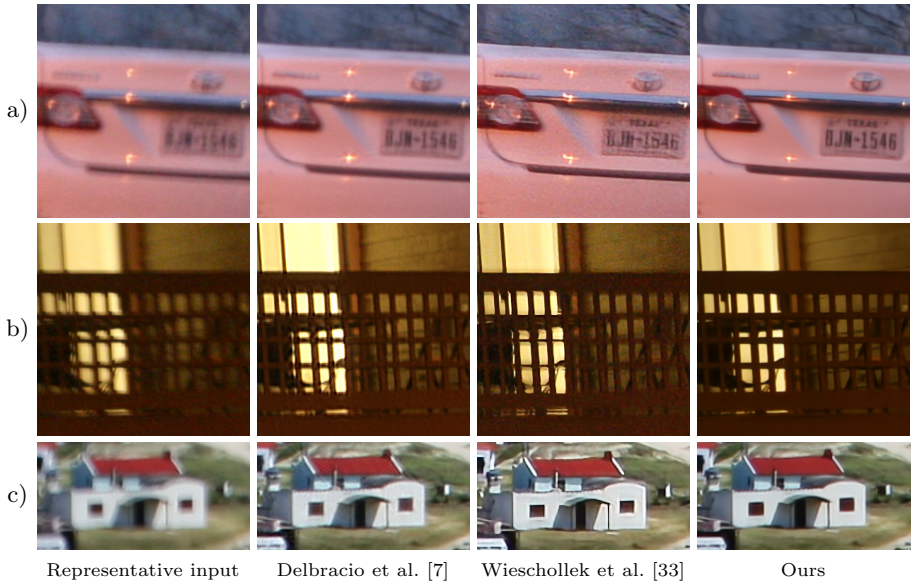



**Fig. 6.** Representative results on the dataset of Delbracio et al. [7], for methods of Delbracio et al. [7] (FBA), Wieschollek et al. [33] (RDN), and our method. Overall, our result is sharper and suffers from fewer artifacts such as oversharpening. Notice in particular the clean detail resolved in the license plate, the properly removed streaks in specular highlights on the car paint, and the artifact-free grid on the balcony railing. For (c), the ground truth is available: SSIM values achieved are 0.9456, 0.8764 and 0.9578 for FBA, RDN and our method, respectively.

## 4 Results

Figures 1 and 5 illustrate a selection of results from our method for a variety of challenging bursts shot under low light conditions with a shaky hand. The bursts were shot with the back camera of an iPhone SE. We use the raw format to bypass the (for us) counterproductive denoising of the camera software. We believe that this dataset is significantly more challenging than the existing ones in literature. The photographs are in their original resolution, which means that the shake kernels are relatively large and the noise has not been averaged down. We avoided including significantly lucky images, which might lead to overly optimistic results.

Overall, our method recovers significantly sharper images than any of those in the input burst. The results are largely free of high-frequency noise, and do not exhibit systematic artifacts besides blurriness in ambiguous regions for low frame counts. The method often extracts information that is collectively preserved by the full burst, but arguably not recoverable from any of the individual frames – see for example the text highlighted in Figure 1, or try to count the number of cars on the sidewalk in Figure 5 (top).

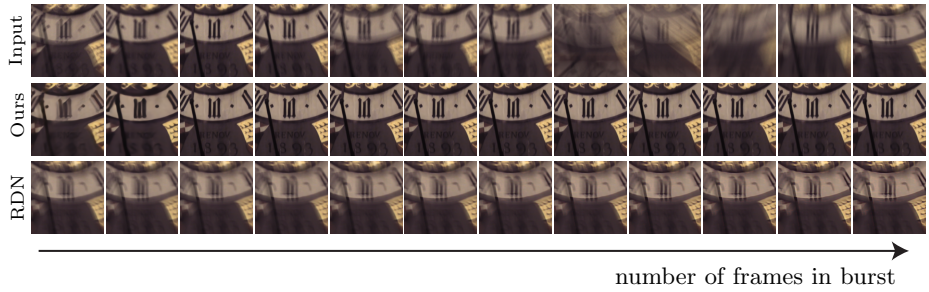



**Fig. 7.** When interpreted as a burst (and aligned by homographies), the standard deconvolution benchmark dataset of Köhler et al. [17] contains a "lucky" sharp frame in the third position. Our method successfully picks it up, and produces a consistently sharp estimate once it gets included in the burst. The poor-quality frames towards the end are also successfully ignored. In contrast, the RDN method of Wieschollek et al. [33] fails to take the lucky frame into account, and focuses on gradually improving the initial estimate. This suggests that a recurrent architecture struggles to give a uniform consideration to all frames it is presented with.

### 4.1 Comparisons and experiments

**Burst deblurring** We compare our method to the state of the art neural burst deblurring method of Wieschollek et al. [33] in Figures 1, 6 and 7. In Figure 6 we use result images provided by the authors, and elsewhere we used their publicly available software implementation. Please refer to the supplemental material for further results on their and other methods [7,34,29,38].

Figure 6 shows comparison results on the dataset of Delbracio et al. [7], which contains various real-world bursts shot with different cameras (we also include their results). While all of the methods provide good results on this dataset, our method consistently reveals more detail, while producing fewer artifacts and exhibiting lower levels of noise. Many of these bursts contain lucky sharp frames and only modest blur and noise. In the more challenging dataset we captured, the method of Wieschollek et al. [33] does not reach a comparable quality, as shown in Figure 1.

Figure 7 shows a result on the dataset of Köhler et al. [17], which contains a mixture of sharp and extremely blurry frames. Our method successfully picks up the lucky frame in the sequence, while the recurrent architecture of Wieschollek et al. [33] fails to properly integrate it into its running estimate. This behavior is confirmed by numerical comparisons to the ground thuth; see the supplemental appendix document for these results and further numerical experiments.

**Video deblurring** While we consider general object motion deblurring to be out of our scope, our method is in principle compatible with the flow-based frame registration scheme of Deep Video Deblurring method of Su et al. [30], as demonstrated in Figure 8. The input is a sequence of five frames where moving objects have been deformed by optical flow to match the center frame (i.e. the

Middle frame | Su et al. [30] | Ours | Su et al. [30] (see Fig 1)

**Fig. 8.** Our method applied on flow-aligned five-frame video segments with moving objects from Su et al. [30]. Our result shows artifacts on the car hood, as it has not been trained to handle the distortions in the input data, but is otherwise sharper (note the tires and the text on the signs). Applied to our data from Figure 1, Su et al. [30] do not reach the same quality (right).

third). Our network is not trained to handle the deformation artifacts, and fails to clean them up, but aside from this our result is sharper. Conversely, when applied to a five-frame sequence from Figure 1 (centered around the sharpest frame), the result from Su et al. [30] is noisier and blurrier than ours.

**Single-image blind deconvolution** To verify that considering the entire burst using our method provides a benefit over simply deblurring the sharpest individual frame, we tested state of the art blind single-image deconvolution methods [19,22] on our data. Figure 9 shows that considering the entire burst with our method results in a significantly better image. As a curiousity, we also tried training our method on solely single-image "bursts"; we reach a comparable or better performance than these dedicated single-image methods on our noisy data, but fall somewhat short in less noisy ones.

**Significance of noise and dynamic range in training data** While we have emphasized the importance of noise modeling, the main benefit of our method is still derived from the permutation invariant architecture. To test this, we trained our method with a naive noise model, simply adding independent normally distributed noise of standard deviation 0.02 on every training input. Figure 10 shows the result: while the output is much noisier, it is still state of the art in terms of detail resolved. Also shown is the effect of omitting the dynamic range expansion scheme.

Input | Kupyn et al. [19] (single-image) | Nah et al. [22] (single-image) | Ours (single-image) | Ours **(full burst)**

**Fig. 9.** Using our method on the full burst produces a significantly better result than deblurring the sharpest frame (left) with state of the art single-image deblurring methods [19,22]. This is to be expected, as the burst contains more information as a whole. When trained exclusively for single-image deblurring, our method also provides comparable or better single-image performance when the input suffers from heavy noise.

## 5 Conclusions

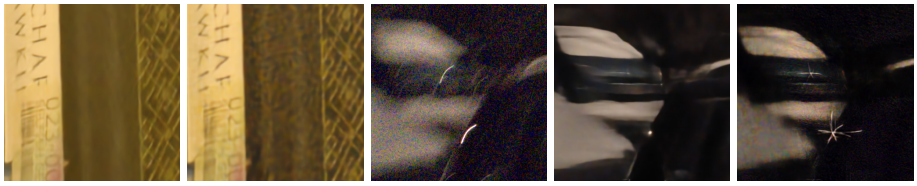
We have presented a method for restoring sharp and noise-free images from bursts of photographs suffering from severe hand tremor and noise. The method reveals accurate image detail and produces pleasing image quality in challenging but realistic datasets that state of the art methods struggle with.

We attribute the success of our method largely to the network architecture that facilitates uniform order-independent handling of the input data, and hope that these ideas will find more widespread use with neural networks. A wide array of interesting problems have the character of fusing together evidence that is scattered in a loosely structured set of observations; one need only think of countless problems that are classically approached by stacking together likelihood terms corresponding to measurement data. Our results also indicate that image restoration methods targeting low-end imaging devices or low-light photography can benefit from considering more complex noise and image degradation models.

**Acknowledgements.** This work was supported by Toyota Research Institute.

## References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng,



a) b) c) d) e)

**Fig. 10.** Impact of noise and dynamic range expansion during training. (a) Our result for dataset of Figure 1 when trained with our full noise model. (b) When trained with a simple non-correlated noise model, the method still exhibits state of the art burst deblurring performance, but leaves in more mid-frequency noise. (c) Representative frame from an extremely degraded burst. (d) Result from our full model. (e) Trained with a simple noise model and no dynamic range expansion, the method underestimates the intensity of noisy dark regions, and fails to concentrate the streaks into a point.

X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`

2. Babacan, S.D., Molina, R., Do, M.N., Katsaggelos, A.K.: Bayesian blind deconvolution with general sparse image priors. In: Proceedings of the 12th European Conference on Computer Vision - Volume Part VI. pp. 341–355. ECCV'12, Springer-Verlag, Berlin, Heidelberg (2012)
3. Cai, J.F., Ji, H., Liu, C., Shen, Z.: Blind motion deblurring using multiple images. J. Comput. Phys. **228**(14), 5057–5071 (Aug 2009)
4. Chakrabarti, A.: A neural approach to blind motion deblurring. In: ECCV (2016)
5. Chen, H., Gu, J., Gallo, O., Liu, M., Veeraraghavan, A., Kautz, J.: Reblur2deblur: Deblurring videos via self-supervised learning. In: 2018 IEEE International Conference on Computational Photography, ICCP 2018, Pittsburgh, PA, USA, May 4-6, 2018. pp. 1–9. IEEE Computer Society (2018)
6. Clevert, D., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). In: ICLR (2016)
7. Delbracio, M., Sapiro, G.: Removing camera shake via weighted fourier burst accumulation. IEEE Transactions on Image Processing **24**(11), 3293–3307 (Nov 2015)
8. Edwards, H., Storkey, A.J.: Towards a neural statistician. In: ICLR (2017)
9. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE Trans. Pattern Anal. Mach. Intell. **30**(10), 1858–1865 (2008)
10. 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 (Jul 2006)
11. Godard, C., Matzen, K., Uyttendaele, M.: Deep burst denoising. CoRR **abs/1712.05790** (2017)
12. Hasinoff, S.W., Sharlet, D., Geiss, R., Adams, A., Barron, J.T., Kainz, F., Chen, J., Levoy, M.: Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Transactions on Graphics (Proc. SIGGRAPH Asia) **35**(6) (2016)
13. Herzig, R., Raboh, M., Chechik, G., Berant, J., Globerson, A.: Mapping images to scene graphs with permutation-invariant structured prediction. CoRR **abs/1802.05451** (2018)

14. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)
15. Kim, T.H., Lee, K.M., Schölkopf, B., Hirsch, M.: Online video deblurring via dynamic temporal blending network. In: Proceedings IEEE International Conference on Computer Vision (ICCV). pp. 4038–4047. IEEE, Piscataway, NJ, USA (Oct 2017)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
17. Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision – ECCV 2012. pp. 27–40. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
18. Korshunova, I., Degrave, J., Huszár, F., Gal, Y., Gretton, A., Dambre, J.: A Generative Deep Recurrent Model for Exchangeable Data. ArXiv e-prints (Feb 2018)
19. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. CVPR (2018)
20. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding blind deconvolution algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2354–2367 (Dec 2011)
21. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. CVPR (2018)
22. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
23. Noroozi, M., Chandramouli, P., Favaro, P.: Motion deblurring in the wild. In: Roth, V., Vetter, T. (eds.) Pattern Recognition - 39th German Conference, GCPR 2017, Basel, Switzerland, September 12-15, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10496, pp. 65–77. Springer (2017)
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 77–85. IEEE Computer Society (2017)
25. Ronneberger, O., P.Fischer, Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). LNCS, vol. 9351, pp. 234–241. Springer (2015)
26. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015)
27. Salimans, T., Kingma, D.P.: Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In: NIPS (2016)
28. 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)
29. Sroubek, F., Milanfar, P.: Robust multichannel blind deconvolution via fast alternating minimization. IEEE Transactions on Image Processing **21**(4), 1687–1700 (April 2012)
30. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1279–1288 (2017)

31. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 769–777 (2015)
32. Wang, R., Tao, D.: Recent progress in image deblurring. CoRR **abs/1409.6838** (2014)
33. Wieschollek, P., Hirsch, M., Schölkopf, B., Lensch, H.: Learning blind motion deblurring. In: IEEE International Conference on Computer Vision (ICCV 2017). pp. 231–240 (2017)
34. Wieschollek, P., Schölkopf, B., Lensch, H.P.A., Hirsch, M.: End-to-end learning for image burst deblurring. In: Lai, S.H., Lepetit, V., Nishino, K., Sato, Y. (eds.) Computer Vision – ACCV 2016. pp. 35–51. Springer International Publishing, Cham (2017)
35. Xu, X., Pan, J., Zhang, Y.J., Wu, Y.: Motion blur kernel estimation via deep learning. IEEE Transactions on Image Processing **27**, 194–205 (2017)
36. Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017. pp. 3394–3404 (2017)
37. Zhang, H., Carin, L.: Multi-shot imaging: Joint alignment, deblurring, and resolution-enhancement. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2925–2932 (June 2014)
38. Zhang, H., Wipf, D., Zhang, Y.: Multi-observation blind deconvolution with an adaptive sparse prior. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8), 1628–1643 (Aug 2014)
39. Zhang, H., Yang, J.: Intra-frame deblurring by leveraging inter-frame camera motion. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4036–4044 (June 2015)