Deep Burst Denoising

Clément Godard[1,*] Kevin Matzen[2] Matt Uyttendaele[2]

[1]University College London [2]Facebook

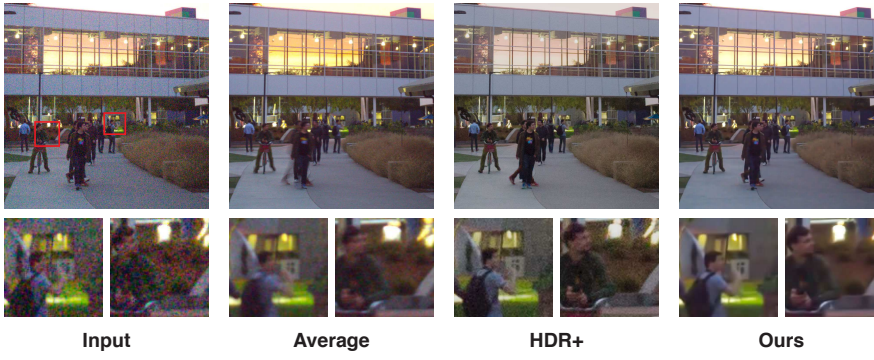


Fig. 1: **Denoising on a real raw burst from [19].** Our method is able to perform high levels of denoising on low-light bursts while maintaining details.

Abstract. Noise is an inherent issue of low-light image capture, which is worsened on mobile devices due to their narrow apertures and small sensors. One strategy for mitigating noise in low-light situations is to increase the shutter time, allowing each photosite to integrate more light and decrease noise variance. However, there are two downsides of long exposures: (a) bright regions can exceed the sensor range, and (b) camera and scene motion will cause blur. Another way of gathering more light is to capture multiple short (thus noisy) frames in a burst and intelligently integrate the content, thus avoiding the above downsides. In this paper, we use the burst-capture strategy and implement the intelligent integration via a recurrent fully convolutional deep neural net (CNN). We build our novel, multi-frame architecture to be a simple addition to any single frame denoising model. The resulting architecture denoises all frames in a sequence of arbitrary length. We show that it achieves state of the art denoising results on our burst dataset, improving on the best published multi-frame techniques, such as VBM4D and FlexISP. Finally, we explore other applications of multi-frame image enhancement and show that our CNN architecture generalizes well to image super-resolution.

* This work was done during an internship at Facebook.

1 Introduction

Noise reduction is one of the most important problems to solve in the design of an imaging pipeline. The most straight-forward solution is to collect as much light as possible when taking a photograph. This can be addressed in camera hardware through the use of a large aperture lens, sensors with large photosites, and high quality A/D conversion. However, relative to larger standalone cameras, e.g. a DSLR, modern smartphone cameras have compromised on each of these hardware elements. This makes noise much more of a problem in smartphone capture.

Another way to collect more light is to use a longer shutter time, allowing each photosite on the sensor to integrate light over a longer period of time. This is commonly done by placing the camera on a tripod. The tripod is necessary as any motion of the camera will cause the collected light to blur across multiple photosites. This technique is limited though. First, any moving objects in the scene and residual camera motion will cause blur in the resulting photo. Second, the shutter time can only be set for as long as the brightest objects in the scene do not saturate the electron collecting capacity of a photosite. This means that for high dynamic range scenes, the darkest regions of the image may still exhibit significant noise while the brightest ones might staturate.

In our method we also collect light over a longer period of time, by capturing a burst of photos. Burst photography addresses many of the issues above (a) it is available on inexpensive hardware, (b) it can capture moving subjects, and (c) it is less likely to suffer from blown-out highlights. In using a burst we make the design choice of leveraging a computational process to integrate light instead of a hardware process, such as in [29] and [19]. In other words, we turn to computational photography.

Our computational process runs in several steps. First, the burst is stabilized by finding a homography for each frame that geometrically registers it to a common reference. Second, we employ a fully convolutional deep neural network (CNN) to denoise each frame individually. Third, we extend the CNN with a parallel recurrent network that integrates the information of all frames in the burst.

The paper presents our work as follows. In section 2 we review previous single-frame and multi-frame denoising techniques. We also look at super-resolution, which can leverage multi-frame information. In section 3 we describe our recurrent network in detail and discuss training. In order to compare against previous work, the network is trained on simulated Gaussian noise. We also show that our solution works well when trained on Poisson distributed noise which is typical of a real-world imaging pipeline [18]. In section 4, we show significant increase in reconstruction quality on burst sequences in comparison to state of the art single-frame denoising and performance on par or better than recent state of the art multi-frame denoising methods. In addition we demonstrate that burst capture coupled with our recurrent network architecture generalizes well to super-resolution.

In summary our main contributions are:

- We introduce a recurrent architecture which is a simple yet effective extension to single-frame denoising models,
- Demonstrate that bursts provide a large improvement over the best deep learning based single-frame denoising techniques,

– Show that our model achieves performance on par with or better than recent state of the art multi-frame denoising methods, and
– Demonstrate that our recurrent architecture generalizes well by applying it to super-resolution.

2 Related work

This work addresses a variety of inverse problems, all of which can be formulated as consisting of (1) a target "restored" image, (2) a temporally-ordered set or "burst" of images, each of which is a corrupted observation of the target image, and (3) a function mapping the burst of images to the restored target. Such tasks include denoising and super-resolution. Our goal is to craft this function, either through domain knowledge or through a data-driven approach, to solve these multi-image restoration problems.

Denoising

Data-driven single-image denoising research dates back to work that leverages block-level statistics within a single image. One of the earliest works of this nature is Non-Local Means [3], a method for taking a weighted average of blocks within an image based on similarity to a reference block. Dabov, *et al*. [9] extend this concept of block-level filtering with a novel 3D filtering formulation. This algorithm, BM3D, is the de facto method by which all other single-image methods are compared to today.

Learning-based methods have proliferated in the last few years. These methods often make use of neural networks that are purely feed-forward [44,4,49,25,15,1,50], recurrent [45], or a hybrid of the two [7]. Methods such as Field of Experts [39] have been shown to be successful in modeling natural image statistics for tasks such as denoising and inpainting with contrastive divergence. Moreover, related tasks such as demosaicing and denoising have shown to benefit from joint formulations when posed in a learning framework [15]. The recent work of [5] applied a recurrent architecture in the context of denoising ray-traced sequenced, and finally [6] used a simple fully connected RNN for video denoising which, while failing to beat VBM4D [33,32], proved the feasibility of using RNNs for video denoising.

Multi-image variants of denoising methods exist and often focus on the best ways to align and combine images. Tico [41] returns to a block-based paradigm, but this time, blocks "within" and "across" images in a burst can be used to produce a denoised estimate. VBM3D [8] and VBM4D [33,32] provide extensions on top of the existing BM3D framework. Liu, *et al*. [29] showed how similar denoising performance in terms of PSNR could be obtained in one tenth the time of VBM3D and one one-hundredth the time of VBM4D using a novel "homography flow" alignment scheme along with a "consistent pixel" compositing operator. Systems such as FlexISP [22] and ProxImaL [21] offer end-to-end formulations of the entire image processing pipeline, including demosaicing, alignment, deblurring, etc., which can be solved jointly through efficient optimization.

We in turn also make use of a deep model and base our CNN architecture on current state of the art single-frame methods [36,49,27].

Super-Resolution

Super-resolution is the task of taking one or more images of a fixed resolution as input and producing a fused or hallucinated image of higher resolution as output.

Nasrollahi, *et al*. [35] offers a comprehensive survey of single-image super-resolution methods and Yang, *et al*. [46] offers a benchmark and evaluation of several methods. Glasner, *et al*. [16] show that single images can be super-resolved without any need of an external database or prior by exploiting block-level statistics "within" the single image. Other methods make use of sparse image statistics [47]. Borman, *et al*. offers a survey of multi-image methods [2]. Farsiu, *et al*. [13] offers a fast and robust method for solving the multi-image super-resolution problem. More recently convolutional networks have shown very good results in single image super-resolution with the works of Dong *et al*. [11] and the state of the art Ledig *et al*. [27].

Our single-frame architecture takes inspiration by recent deep super-resolution models such as [27].

2.1 Neural Architectures

It is worthwhile taking note that while image restoration approaches have often been learning-based in recent years, there's also great diversity in how those learning problems are modeled. In particular, neural network-based approaches have experienced a gradual progression in architectural sophistication over time.

In the work of Dong, *et al*. [10], a single, feed-forward CNN is used to super-resolve an input image. This is a natural design as it leveraged what was then new advancements in discriminatively-trained neural networks designed for classification and applied them to a regression task. The next step in architecture evolution was to use Recurrent Neural Networks, or RNNs, in place of the convolutional layers of the previous design. The use of one or more RNNs in a network design can both be used to increase the effective depth and thus receptive field in a single-image network [45] or to integrate observations across many frames in a multi-image network. Our work makes use of this latter principle.

While the introduction of RNNs led to network architectures with more effective depth and thus a larger receptive field with more context, the success of skip connections in classification networks [20] and segmentation networks [40,37] motivated their use in restoration networks. The work of Remez, *et al*. [36] illustrates this principle by computing additive noise predictions from each level of the network, which then sum to form the final noise prediction.

We also make use of this concept, but rather than use skip connections directly, we extract activations from each level of our network which are then fed into corresponding RNNs for integration across all frames of a burst sequence.

3 Method

In this section we first identify a number of interesting goals we would like a multi-frame architecture to meet and then describe our method and how it achieves such goals.

3.1 Goals

Our goal is to derive a method which, given a sequence of noisy images produces a denoised sequence. We identified desirable properties, that a multi-frame denoising technique should satisfy:

1. **Work for single-frame denoising.** A corollary to the first criterion is that our method should be competitive for the single-frame case.
2. **Generalize to any number of frames.** A single model should produce competitive results for any number of frames that it is given.
3. **Denoise the entire sequence.** Rather than simply denoise a single reference frame, as is the goal in most prior work, we aim to denoise the entire sequence, putting our goal closer to video denoising.
4. **Be robust to motion.** Most real-world burst capture scenarios will exhibit both camera and scene motion.
5. **Be temporally coherent.** Denoising the entire sequence requires that we do not introduce flickering in the result.
6. **Generalize to a variety of image restoration tasks.** As discussed in Section 2, tasks such as super-resolution can benefit from multi-frame methods, albeit, trained on different data.

In the remainder of this section we will first describe a single-frame denoising model that produces competitive results with current state of the art models. Then we will discuss how we extend this model to accommodate an arbitrary number of frames for multi-frame denoising and how it meets each of our goals.

3.2 Single frame denoising

We treat image denoising as a structured prediction problem, where the network is tasked with regressing a pixel-aligned denoised image $\tilde{I}_s = f_s(N, \theta_s)$ from noisy image N, given the model parameters θ_s. Following [51] we train the network by minimizing the L1 distance between the predicted output and the ground-truth target image, I.

$$E_{\text{SFD}} = |I - f_s(N, \theta_s)| \tag{1}$$

To be competitive in the single-frame denoising scenario, and to meet our 1st goal, we take inspiration from the state of the art to derive an initial network architecture. Several existing architectures [49,36,27] consist of the same base design: a fully convolutional architecture consisting of L layers with C channels each.

We follow suit and choose this simple architecture to be our single frame denoising (SFD) baseline, with $L = 8$, $C = 64$, 3×3 convolutions and ReLU [31] activation functions, except on the last layer.

3.3 Multi-frame denoising

Following goals 1-3, our model should be competitive in the single-frame case while being able to denoise the entire input sequence. In other words, using a set of noisy

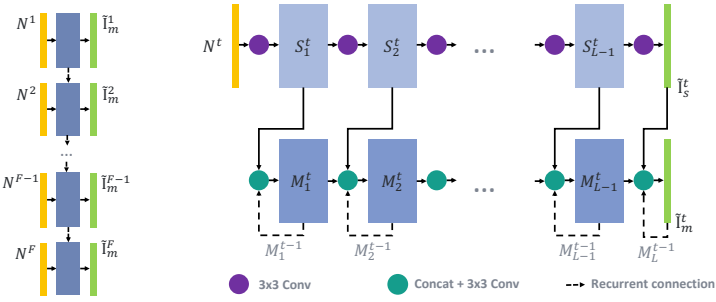


Fig. 2: **Global recurrent architecture (left).** Our model takes as input F noisy frames N^t and predicts F clean frames $\tilde{I}_m^t$. **Local recurrent architecture (right).** The top part of our model is a single-frame denoiser (SFD, in light blue): it takes as input a noisy image N^t and regresses a clean image $\tilde{I}_s^t$, its features S_i^t are fed to the multi-frame denoiser (MFD, in darker blue) which also makes use of recurrent connections from the previous state (dotted lines) to output a clean image $\tilde{I}_m^t$.

images as input, forming the sequence $\{N^t\}$, we want to regress a denoised version of each noisy frame, $\tilde{I}_m^t = f_m^t(\{N^t\}, \theta_m)$, given the model parameters θ_m. Formally, our complete training objective is:

$$\begin{aligned} E &= \sum_t^F E_{\text{SFD}}^t + E_{\text{MFD}}^t \\ &= \sum_t^F |I^t - f_s(N^t, \theta_s)| + |I^t - f_m^t(\{N^t\}, \theta_m)| \end{aligned} \tag{2}$$

A natural approach, which is already popular in the natural language and audio processing literature [48], is to process temporal data with recurrent neural network (RNN) modules [23]. RNNs operate on sequences and maintain an internal state which is combined with the input at each time step. As can be seen in Figure 2, our model makes use of recurrent connections to aggregate activations produced by our SFD network for each frame. This satisfies our first goal as it allows for an arbitrary input sequence length.

Unlike [5] and [43] which use a single-track network design, we use a two track network architecture with the top track dedicated to SFD and the bottom track dedicated to fusing those results into a final prediction for MFD. This two track design decouples decoupling per-frame feature extraction from multi-frame aggregation, enabling the possibility for pre-training a network rapidly using only single-frame data. In practice, we found that this pre-training not only accelerates the learning process, but also produces significantly better results in terms of PSNR than when we train the entire MFD from scratch. The core intuition is that by first learning good features for SFD, we put the network in a good state for learning how to aggregate those features across observations.

It is also important to note that the RNNs are connected in such a way as to permit the aggregation of features in several different ways. Temporal connections within the

RNNs help aggregate information "across" frames, but lateral connections "within" the MFD track permit the aggregation of information at different physical scales and at different levels of abstraction.

4 Implementation and Results

We evaluate our method with the goals from Section 3 in mind, and examine: single-image denoising (goal 1), multi-frame denoising (goals 2-5), and multi-frame super-resolution (goal 6). In Section 4.5 we compare different single-frame denoising approaches, showing that quality is plateauing despite the use of deep models and that our simple single-frame denoiser is competitive with state-of-the-art. In Section 4.6 we show that our method significantly outperforms the reference state of the art video denoising method VBM4D [32]. Finally in Section 4.7 we compare our method to the state of the art burst denoising methods HDR+ [19], FlexISP [22] and ProximaL [21] on the FlexISP dataset.

4.1 Data

We trained all the networks in our evaluation using a dataset consisting of Apple Live Photos. Live Photos are burst sequences captured by Apple iPhone 6S and above[1]. This dataset is very representative as it captures what mobile phone users often photograph, and exhibits a wide range of scenes and motions. Approximately 73k public sequences were scraped from a social media website with a resolution of 360×480. We apply a burst stabilizer to each sequence, resulting in approximately 54.5k sequences successfully stabilized. In Section 4.2 we describe our stabilization procedure in more detail. 50k sequences were used for training with an additional 3.5k reserved for validation and 1k reserved for testing.

4.2 Stabilization

We implemented burst sequence stabilization using OpenCV[2]. In particular, we use a Lucas-Kanade tracker [30] to find correspondences between successive frames and then a rotation-only motion model and a static focal length guess to arrive at a homography for each frame. We warp all frames of a sequence back into a reference frame's pose then crop and scale the sequence to maintain the original size and aspect ratio, but with the region of interest contained entirely within the valid regions of the warp. The stabilized sequences still exhibit some residual motion, either through moving objects or people, or through camera motion which cannot be represented by a homography. This residual motion forces the network to adapt to non static scenes. Stabilization and training on any residual motion makes our system robust to motion, achieving our 4^{th} goal. As we show in supplementary material, stabilization improves the final results, but is not a requirement.

[1] https://support.apple.com/en-us/HT207310
[2] https://opencv.org/

4.3 Training details

We implemented the neural network from Section 3 using the Caffe2 framework[3]. Each model was trained using 4 Tesla M40 GPUs. As described in Section 3, training took place in two stages. First a single-frame model was trained. This model used a batch size of 128 and was trained for 500 epochs in approximately 5 hours. Using this single-frame model as initialization for the multi-frame (8-frame) model, we continue training with a batch size of 32 to accommodate the increased size of the multi-frame model. This second stage was trained for 125 epochs in approximately 20 hours.

We used Adam [26] with a learning rate of 10^{-4} which decays to zero following a square root law. We trained on 64×64 crops with random flips. Finally, we train the multi-frame model using back-propagation through time [42].

4.4 Noise modelling

In order to make comparison possible with previous methods, such as VBM4D, we first evaluate our architecture using additive white Gaussian noise with $\sigma = 15, 25, 50$ and 75. Additionally, to train a denoiser for real burst sequences, we implement a simulated camera processing pipeline. First real world sensor noise is generated following [14]. Separate models are trained using Poisson noise, labelled *a* in [14], with intensity ranging from 0.001 to 0.01. We simulate a Bayer mosaic on a linearized version of our training data and add the Poisson noise to this. Next we reconstruct an RGB image using bilinear interpolation followed by conversion to sRGB and clipping. In both Gaussian and Poisson cases we add synthetic noise *before* stabilization. While it is possible to obtain a single "blind" model by training on multiple noise levels at once [50], it typically results in a small loss in accuracy. We thus follow the protocol established by [49,36] and train a separate model for each noise level, without loss of generality.

	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 75$
BM3D	31.10	28.57	25.62	24.20
TNRD	31.41	28.91	25.95	-
DenoiseNet [36]	31.44	29.04	26.06	24.61
DnCNN [49]	**31.73**	**29.23**	**26.23**	-
Ours single-frame 8L	31.15	28.63	25.65	24.11
Ours single-frame 20L	31.29	28.82	26.02	24.43

Table 1: **Single frame additive white Gaussian noise denoising comparison on BSD68 (PSNR).** Our baseline SFD models match BM3D at 8 layers and get close to both DnCNN and DenoiseNet at 20 layers.

4.5 Single frame denoising

Here, we compare our baseline single frame denoiser with current state of the art methods on additive white Gaussian noise. This shows that single-frame denoising has reached

[3] https://caffe2.ai/

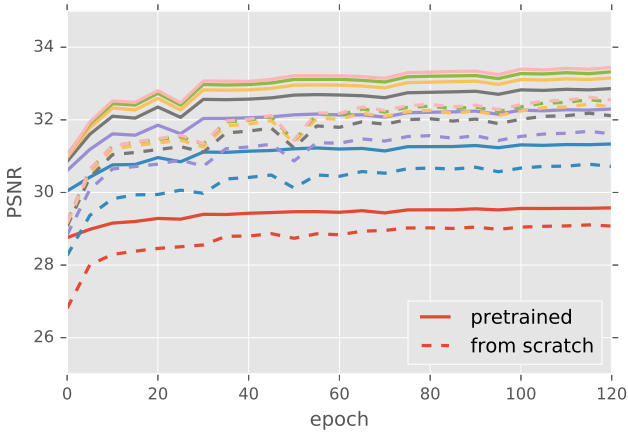


Fig. 3: **Effect of pre-training on multi-frame denoising with Gaussian noise $\sigma = 50$.** Each color corresponds to the average PSNR of the frames in a sequence: 1^{st} (red), 2^{nd} (blue), 3^{rd} (purple) 4^{th} (grey), 5^{th} (yellow) and 6^{th} (pink). As we can see the pre-trained model shows a constant lead of 0.5dB over the model trained from scratch, and reaches a stable state much quicker.

a point of diminishing returns, where significant model complexity is needed improve quality by more than $\sim 0.2dB$.

We compare our own SFD, which is composed of 8 layers, with two 20 layer networks: DenoiseNet (2017) [36] and DnCNN (2017) [49]. For the sake of comparison, we also include a 20 layer version of our SFD. All models were trained for 2000 epochs on 8000 images from the PASCAL VOC2010 [12] using the training split from [36]. We also compare with traditional approaches, such as BM3D (2009) [9] and TNRD (2015) [7].

All models were tested on BSD68 [39], a set of 68 natural images from the Berkeley Segmentation Dataset [34]. In Table 1, we can see diminishing returns in single frame denoising PSNR over the years despite the use of deep neural networks, which confirms what Levin, *et al.* describe in [28]. We can see that our simpler SFD 20 layers model only slightly underperforms both DenoiseNet and DnCNN by $\sim 0.2dB$. However, as we show in the following section, the PSNR gains brought by multi-frame processing vastly outshine fractional single frame PSNR improvements.

4.6 Burst denoising

We evaluate our method on a held-out test set of Live Photos with synthetic additive white Gaussian noise added. In Table 3, we compare our architecture with single frame models as well as the multi-frame method VBM4D [33,32]. We show qualitative results with $\sigma = 50$ in Figure 5.

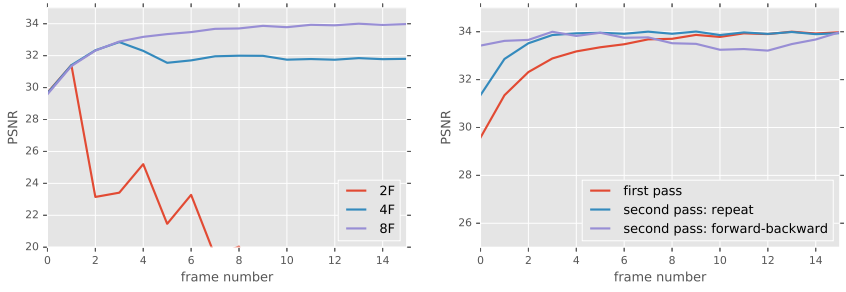
	C2F	C4F	C8F	Ours 4L	**Ours 8L**	Ours 12L	Ours 16L	Ours 20L	Ours *nostab*
PSNR	30.89	31.83	32.15	33.01	**33.62**	33.80	33.35	33.48	32.60

Table 2: **Ablation study on the Live Photos test sequences with additive white Gaussian Noise of** $\sigma = 50$. All models were trained on 8 frame sequences. C2F, C4F and C8F represent **Concat** models which were trained on respectively 2, 4, and 8 concatenated frames as input. Ours *nostab* was trained and tested on the unstabilized sequences.

Ablation study We now evaluate our architecture choices, where we compare our full model, with 8 layers and trained on sequences of 8 frames with other variants.

Concat We first compare our method with a naive multi-frame denoising approach, dubbed **Concat**, where the input consists of n concatenated frames to a single pass denoiser. We evaluated this architecture with $L = 20$ as well as $n = 2, 4$ and 8. As we can see in Table 2 this model performs significantly worse than our model.

Number of layers We also evaluate the impact of the depth of the network by experimenting with $N = 4, 8, 12, 16$ and 20. As can be seen in Figure 2, the 16 and 20 layers network fail to surpass both the 8 and 12 layers after 125 epochs of training, likely because training becomes unstable with increased depth and parameter count [20]. While the 12 layers network shows a marginal 0.18dB increase over the 8 layer model, we decided to go with the latter as we did not think that the modest increase in PSNR was worth the 50% increase in both memory and computation time.



(a) Tranining sequence length (b) Frame ordering

Fig. 4: (a) **Impact of the length** F **of training sequences at test time.** We test 3 models which were trained with $F = 2, 4$ and 8 on 16 frames-long test sequences. (b) **Effect of frame ordering at test time.** We can see the burn-in period on the first pass (red) as well as on the repeat pass. Feeding the sequence forward, then backward, mostly alleviates this problem.

Length of training sequences Perhaps the most surprising result we encountered during training our recurrent model, was the importance of the number of frames in the training sequences. In Figure 4a, we show that models trained on sequences of both 2 and 4 frames fail to generalize beyond their training length sequence. Only models trained

with 8 frames were able to generalize to longer sequences at test time, and as we can see still denoise beyond 8 frames.

Pre-training One of the main advantages of using a two-track network is that we can first train the SFD track independently. As just mentioned, a sequence length of 8 is required to ensure generalization to longer sequences, which makes the training of the full model much slower than training the single-frame pass. As we show in Figure 3, pre-training makes training the MFD significantly faster.

Frame ordering Due to its recurrent nature, our network exhibits a period of burn-in, where the first frames are being denoised to a lesser extent than the later ones. In order to denoise an entire sequence to a high quality level, we explored different options for frame ordering. As we show in Figure 4b, by feeding the sequence twice to the network, we are able to achieve a comparable denoising quality on all frames thus obtaining a higher average PSNR. We propose two variants, either **repeat** the sequence in the same order or reverse it the second time (named **forward-backward**). As we show in Figure 4b, the forward-backward schedule does not suffer from burn-in and remains temporally coherenent, meeting our 5$^{\text{th}}$ goal. We use forward-backward for all our experiments.

Fig. 5: **Multi-frame Gaussian denoising on stabilized Live Photo test data with $\sigma = 50$.** We can see that our MFD produces a significantly sharper image than both our SFD and VBM4D.

4.7 Existing datasets

Here we evaluate our method on existing datasets, showing generization and allowing us to compare with other state-of-the-art denoising approaches. In Figures 1 and 7 we demonstrate that our method is capable of denoising real sequences. This evaluation was performed on real noisy bursts from HDR+ [19]. Please see our supplementary material for more results.

In Figure 6 we show the results of our method on the FlexISP dataset, comparing it with FlexISP, HDR+ and ProximaL on the FlexISP. The dataset consists of 4 noisy sequences: 2 synthetic (FLICKR DOLL and KODAK FENCE) and 2 real ones (DARKPAINTCANS and LIVINGROOM). The synthetic sequences were generated by randomly warping the input images and introducing: (for FLICKR DOLL) additive and multiplicative white Gaussian noise with $\sigma = 25.5$, and (for KODAK FENCE) additive with Gaussian noise of $\sigma = 12$ while simulating a Bayer filter. We trained a model for each synthetic scene, by replicating by replicating the corresponding noise conditions

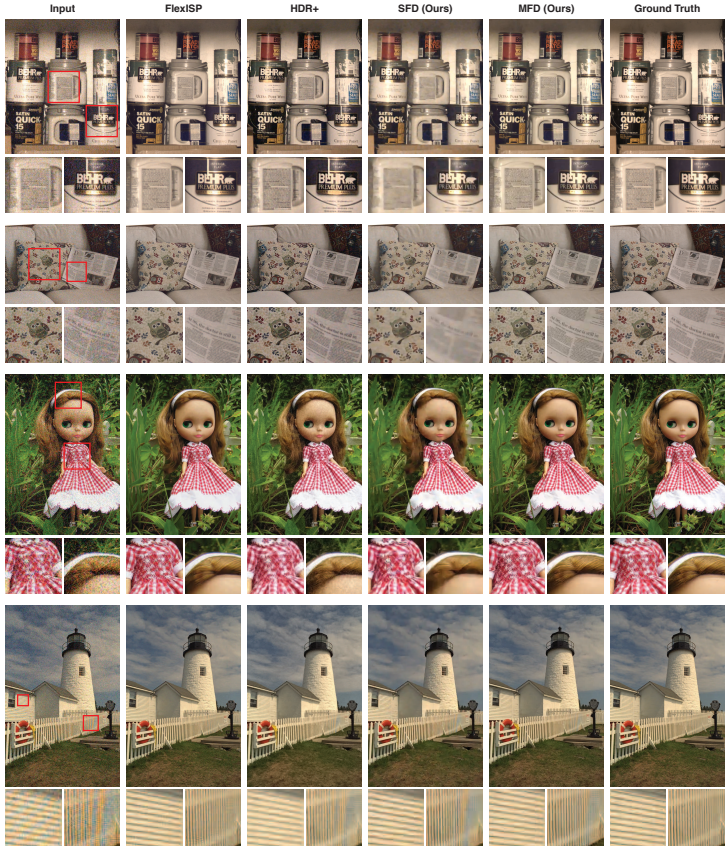


Fig. 6: **Denoising results on two real and two synthetic bursts on the FlexISP dataset** [22]. From top to bottom: DARKPAINTCANS, LIVINGROOM, FLICKR DOLL and KODAK FENCE. Our recurrent model is able to match the quality of FlexISP on FLICKR DOLL and beats it by 0.5dB on KODAK FENCE.

	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 75$
BM3D	35.67	32.92	29.41	27.40
DnCNN	35.84	32.93	29.13	27.06
DenoiseNet	35.91	33.17	29.56	27.49
VBM4D	36.42	33.41	29.14	26.60
Ours	**39.23**	**36.87**	**33.62**	**31.44**

	FLICKR DOLL	KODAK FENCE
BM3D	25.47	31.09
VBM3D	27.48	31.60
FlexISP	29.41	34.44
ProximaL	**30.23**	-
Ours	29.39	**34.98**

Table 3: **Multi-frame denoising comparison on Live Photo sequences (left) and the FlexISP sequences (right)**. Average PSNR for all frames on 1000 test 16-frames sequences with additive white Gaussian noise. **Multi-frame denoising comparison on the FlexISP images (right).** Best results are in bold. The thick line separates single frame methods and multi-frame ones.

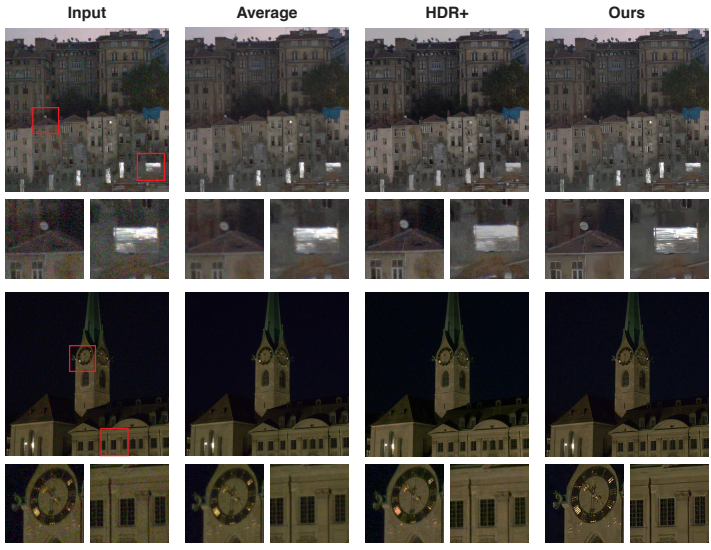


Fig. 7: **Denoising results on two real bursts on the HDR+ dataset** [19]. Our method produces a high level of denoising while keeping sharp details and maintaining information in highlights.

on our Live Photos dataset. To match the evaluation of previous work, we used only the first 8 frames from each sequence for denoising.

Table 3 shows that our method matches FlexISP on FLICKR DOLL and achieves a significant advantage of 0.5dB over FlexISP KODAK FENCE. Interestingly, our method reaches a higher PSNR than FlexISP, despite showing some slight demosiacing artifacts on the fence (see In Figure 6). This is likely due to the absence of high frequency demosaicing artifacts in our training data and would probably be fixed by generating training data following the same protocol as the test data.

Unfortunately, it is difficult to compare a thoroughly with ProximaL, as the publicly implementation does not have code for their experiments. We attempted to reimplement burst denoising using their publicly available framework, but were unable to produce stable results. As ProximaL only shows denoising results on FLICKR DOLL, this limits us to a less comprehensive comparison on only one scene, where our method falls short.

Like HDR+, we do not report quantitative results on the real scenes (DARKPAINTCANS and LIVINGROOM), as we were unable to correct for a color shift between the ground truth long exposure images and the noisy bursts. However, Figure 6 shows that our method is able to recover a lot of details while removing the noise on these bursts.

4.8 Super resolution

To illustrate that our approach generalizes to tasks beyond denoising, and to meet our 6^{th} goal, we trained our model to perform $4\times$ super-resolution, while keeping the rest of the training procedure identical to that of the denoising pipeline. That is, instead of

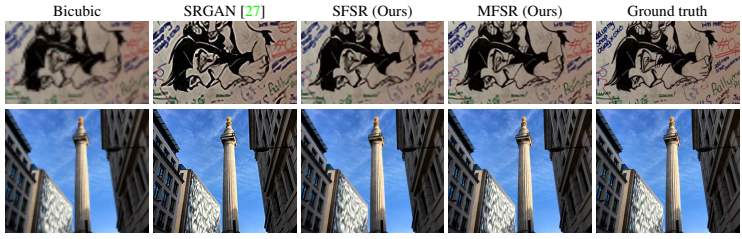


Fig. 8: **Multi-frame $4\times$ super-resolution on stabilized Live Photo test data.** While our single frame model achieves a good upsampling, the increase in sharpness from our multi-frame approach brings a significant quality improvement.

using a burst of noisy images as input, we provide our network with a burst of low-resolution images and task it to provide us with a sharp high-resolution output. To keep the architecture the same, we do not feed low-resolution images as input to the network, but instead remove high-frequency details by first downsampling each input patch $4\times$ and then resize them back to their original size with bilinear interpolation. Figure 8 shows how our multi-frame model is able to recover high-frequency details, such as the crisp contours of the lion and the railing on top of the pillar.

5 Limitations

Our single-frame architecture, based on [36,49,27], makes use of full resolution convolutions. They are however both memory and computationally expensive, and have a small receptive field for a given network depth. Using multiscale architectures, such as a U-Nets [38], could help alleviate both issues, by reducing the computational and memory load, while increasing the receptive field. While not necessary, we trained our network on pre-stabilized sequences, we observed a drop in accuracy on unstabilized sequences, as can be seen in Table 2, as well as instability on longer sequences. It would be interesting to train the network to stabilize the sequence by warping inside the network such as in [24,17]. Finally the low resolution of our training data prevents the model from recoving high frequency details; a higher resolution dataset would likely fix this issue.

6 Conclusion

We have presented a novel deep neural architecture to process burst of images. We improve on a simple single frame architecture by making use of recurrent connections and show that while single-frame models are reaching performance limits, our recurrent architecture vastly outperforms such models for multi-frame data. We carefully designed our method to align with the goals we stated in Section 3.1. As a result, our approach achieves state-of-the-art performance in our Live Photos dataset, and matches or beats existing multi-frame denoisers on challenging existing real-noise datasets.

Acknowledgments We would like to thank Sam Hasinoff and Andrew Adams for the HDR+ dataset, Jan Kautz for the FlexISP dataset and Ross Grishick for the helpful discussions. Finally huge thanks to Peter Hedman for his last minute magic.

References

1. Agostinelli, F., Anderson, M.R., Lee, H.: Adaptive multi-column deep neural networks with application to robust image denoising. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 1493–1501. Curran Associates, Inc. (2013) 3
2. Borman, S., Stevenson, R.L.: Super-resolution from image sequences-a review. In: Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on. pp. 374–378. IEEE (1998) 4
3. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 60–65. IEEE (2005) 3
4. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2392–2399. IEEE (2012) 3
5. Chaitanya, C.R.A., Kaplanyan, A.S., Schied, C., Salvi, M., Lefohn, A., Nowrouzezahrai, D., Aila, T.: Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. ACM Transactions on Graphics (TOG) **36**(4), 98 (2017) 3, 6
6. Chen, X., Song, L., Yang, X.: Deep rnns for video denoising. In: Applications of Digital Image Processing XXXIX. vol. 9971, p. 99711T. International Society for Optics and Photonics (2016) 3
7. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1256–1272 (2017) 3, 9
8. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing **16**(8), 2080–2095 (2007) 3
9. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Bm3d image denoising with shape-adaptive principal component analysis. In: SPARS'09-Signal Processing with Adaptive Sparse Structured Representations (2009) 3, 9
10. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(2), 295–307 (Feb 2016). https://doi.org/10.1109/TPAMI.2015.2439281 4
11. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2016) 4
12. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (Jun 2010) 9
13. Farsiu, S., Robinson, M.D., Elad, M., Milanfar, P.: Fast and robust multiframe super resolution. IEEE transactions on image processing **13**(10), 1327–1344 (2004) 4
14. Foi, A.: Clipped noisy images: Heteroskedastic modeling and practical denoising. Signal Processing **89**(12), 2609–2629 (2009) 8
15. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Transactions on Graphics (TOG) **35**(6), 191 (2016) 3
16. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: ICCV (2009), http://www.wisdom.weizmann.ac.il/~vision/SingleImageSR.html 4
17. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 14

18. Hasinoff, S.W., Durand, F., Freeman, W.T.: Noise-optimal capture for high dynamic range photography. In: CVPR. pp. 553–560. IEEE Computer Society (2010), http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#HasinoffDF10 2
19. 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 (TOG) **35**(6), 192 (2016) 1, 2, 7, 11, 13
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (June 2016). https://doi.org/10.1109/CVPR.2016.90 4, 10
21. Heide, F., Diamond, S., Nießner, M., Ragan-Kelley, J., Heidrich, W., Wetzstein, G.: Proximal: Efficient image optimization using proximal algorithms. ACM Transactions on Graphics (TOG) **35**(4), 84 (2016) 3, 7
22. Heide, F., Steinberger, M., Tsai, Y.T., Rouf, M., Pajak, D., Reddy, D., Gallo, O., Liu, J., Heidrich, W., Egiazarian, K., et al.: Flexisp: A flexible camera image processing framework. ACM Transactions on Graphics (TOG) **33**(6), 231 (2014) 3, 7, 12
23. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences **79**(8), 2554–2558 (1982) 6
24. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in Neural Information Processing Systems. pp. 2017–2025 (2015) 14
25. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In: Koller, D., Schurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 21, pp. 769–776. Curran Associates, Inc. (2009), http://papers.nips.cc/paper/3506-natural-image-denoising-with-convolutional-networks.pdf 3
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2014) 8
27. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 3, 4, 5, 14
28. Levin, A., Nadler, B.: Natural image denoising: Optimality and inherent bounds. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 2833–2840. IEEE (2011) 9
29. Liu, Z., Yuan, L., Tang, X., Uyttendaele, M., Sun, J.: Fast burst images denoising. ACM Transactions on Graphics (TOG) **33**(6), 232 (2014) 2, 3
30. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2. pp. 674–679. IJCAI'81, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1981), http://dl.acm.org/citation.cfm?id=1623264.1623280 7
31. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. ICML. vol. 30 (2013) 5
32. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. IEEE Transactions on image processing **21**(9), 3952–3966 (2012) 3, 7, 9
33. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.O.: Video denoising using separable 4d nonlocal spatiotemporal transforms. In: Image Processing: Algorithms and Systems. p. 787003 (2011) 3, 9, 11
34. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. vol. 2, pp. 416–423. IEEE (2001) 9

35. Nasrollahi, K., Moeslund, T.B.: Super-resolution: a comprehensive survey. Machine vision and applications **25**(6), 1423–1468 (2014) 4
36. Remez, T., Litany, O., Giryes, R., Bronstein, A.M.: Deep class aware denoising. arXiv preprint arXiv:1701.01698 (2017) 3, 4, 5, 8, 9, 14
37. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, pp. 234–241. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4$_2$8, https://doi.org/10.1007/978-3-319-24574-4_28 4
38. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241. Springer (2015) 14
39. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 860–867. IEEE (2005) 3, 9
40. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4), 640–651 (April 2017). https://doi.org/10.1109/TPAMI.2016.2572683 4
41. Tico, M.: Multi-frame image denoising and stabilization. In: Signal Processing Conference, 2008 16th European. pp. 1–4. IEEE (2008) 3
42. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. Neural networks **1**(4), 339–356 (1988) 8
43. Wieschollek, P., Hirsch, M., Scholkopf, B., Lensch, H.P.A.: Learning blind motion deblurring. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 6
44. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 341–349. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4686-image-denoising-and-inpainting-with-deep-neural-networks.pdf 3
45. Xinyuan Chen, Li Song, X.Y.: Deep rnns for video denoising. In: Proc.SPIE. vol. 9971, pp. 9971 – 9971 – 10 (2016). https://doi.org/10.1117/12.2239260, http://dx.doi.org/10.1117/12.2239260 3, 4
46. Yang, C.Y., Ma, C., Yang, M.H.: Single-image super-resolution: A benchmark. In: European Conference on Computer Vision. pp. 372–386. Springer (2014) 4
47. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE transactions on image processing **19**(11), 2861–2873 (2010) 4
48. Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923 (2017) 6
49. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**, 3142–3155 (2017) 3, 5, 8, 9, 14
50. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 3, 8
51. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. IEEE Transactions on Computational Imaging **3**(1), 47–57 (2017) 5