Noname manuscript No.
(will be inserted by the editor)

Non-Local Video Denoising by CNN

Axel Davy · Thibaud Ehret · Jean-Michel Morel · Pablo Arias · Gabriele Facciolo

Received: date / Accepted: date

Abstract Non-local patch based methods were until recently state-of-the-art for image denoising but are now outperformed by CNNs. Yet they are still the state of the art for video denoising, as video redundancy is a key factor to attain high denoising performance. The problem is that CNN architectures are hardly compatible with the search for self-similarities. In this work we propose a new and efficient way to feed video self-similarities to a CNN. The non-locality is incorporated into the network via a first non-trainable layer which finds for each patch in the input image its most similar patches in a search region. The central values of these patches are then gathered in a feature vector which is assigned to each image pixel. This information is presented to a CNN which is trained to predict the clean image. We apply the proposed method to image and video denoising. In the case of video, the patches are searched for in a 3D spatio-temporal volume. The proposed method achieves state-of-the-art results.

Keywords Denoising · Video denoising · Non-local · Patch-based methods · CNN

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research. Work partly financed by IDEX Paris-Saclay IDI 2016, ANR-11-IDEX-0003-02, ONR grant N00014-17-1-2552, CNES MISS project, DGA Astrid ANR-17-ASTR-0013-01, DGA ANR-16-DEFA-0004-01, MENRT. This work used HPC resources from the "Mésocentre" computing center of CentraleSup 'elec and ENS Paris-Saclay supported by CNRS and Région Île-de-France.

All authors were with Centre Borelli, ENS Paris-Saclay, CNRS, Université Paris-Saclay, 91190 Gif-sur-Yvette, France
E-mail: axel.davy@ens-paris-saclay.fr

1 Introduction

Advances in image sensor hardware have steadily improved the acquisition quality of image and video cameras. However, a low signal-to-noise ratio is unavoidable in low lighting conditions if the exposure time is limited (for example to avoid motion blur). This results in high levels of noise, which negatively affects the visual quality of the video and hinders its use for many applications. As a consequence, denoising is a crucial component of any camera pipeline. Furthermore, by interpreting denoising algorithms as proximal operators, several inverse problems in image processing can be solved by iteratively applying a denoising algorithm [48]. Hence the need for video denoising algorithms with a low running time.

Literature review on image denoising. Image denoising has a vast literature where a variety of methods have been applied: PDEs and variational methods (including MRF models) [11, 50, 51], transform domain methods [21], non-local (or patch-based) methods [7, 18], multiscale approaches [26], etc. See [35] for a review. In the last two or three years, CNNs have taken over the state of the art. In addition to attaining better results, CNNs are amenable to efficient parallelization on GPUs potentially enabling real-time performance. We can distinguish two types of CNN approaches: *trainable inference networks* and *black box* networks.

In the first type, the architecture mimics the operations performed by a few iterations of optimization algorithms used for MAP inference with MRFs prior models. Some approaches are based on the Field-of-Experts model [50], such as [5, 14, 54]. The architecture of [60] is based on EPLL [66], which models the *a priori* distribution of image patches as a Gaussian mixture

model. Trainable inference networks reflect the operations of an optimization algorithm, which leads in some cases to unusual architectures, and to some restrictions in the network design. For example, in the *trainable reaction diffusion network* (TRDN) of [14] even layers must be an image (i.e. have only one feature). As pointed out in [33] these architectures have strong similarities with the residual networks of [28].

The black-box approaches treat denoising as a standard regression problem. They do not use much of the domain knowledge acquired during decades of research in denoising. In spite of this, these techniques are currently topping the list of state-of-the-art algorithms. The first denoising approaches using neural networks were proposed in the mid and late 2000s. Jain and Seung [31] proposed a five layer CNN with 5×5 filters, with 24 features in the hidden layers and sigmoid activation functions. Burger et al. [10] reported the first state-of-the-art results with a multilayer perceptron trained to denoise 17×17 patches, but with a heavy architecture. More recently, DnCNN [64] obtained impressive results with a far lighter 17 layer deep CNN with 3×3 convolutions, ReLU activations and batch normalization [30]. This work also proposes a blind denoising network that can denoise an image with an unknown noise level $\sigma \in [0, 55]$, and a multi-noise network trained to denoise blindly three types of noise. A faster version of DnCNN, named FFDNet, was proposed in [65], which also allows handling spatially varying noise by adding a noise map $\sigma(x)$ as an additional input. The architectures of DnCNN and FFDnet keep the same image size throughout the network. Other architectures [12, 41, 53] use pooling or strided convolutions to downscale the image, and then up-convolutional layers to upscale it back. Skip connections connect the layers before the pooling with the output of the up-convolution to avoid loss of spatial resolution. Skip connections are used extensively in [57].

Although these architectures produce very good results, for textures formed by repetitive patterns non-local patch-based methods still perform better [10, 64]. Some works have therefore attempted to incorporate the non-local patch similarity into a CNN framework. Qiao *et al.* [46] proposed inference networks derived from the non-local FoE MRF model [56]. This can be seen as a non-local version of the TRDN network of [14]. A different non-local TRDN was introduced by [36]. BM3D-net [63] pre-computes for each pixel a stack of similar patches which are fed into a CNN, which reproduces the operations done by (the first step of) the BM3D algorithm: a linear transformation of the group of patches, a non-linear shrinkage function and a second linear transform (the inverse of the first). The authors train the linear transformations and the shrinkage function. In [16] the authors propose an iterative approach that can be used to reinforce non-locality to any denoiser. Each iteration consists of the application of the denoiser followed by a non-local filtering step using a fixed image (denoised with BM3D) for computing the non-local correspondences. An inconvenience is that the resulting algorithm requires to iterate the denoising network. Trainable non-local modules have been recently proposed by using differentiable relaxations of the 1 nearest neighbors [37] and k nearest neighbors [44] selection rules.

Literature review on video denoising. CNNs have been successfully applied to several video processing tasks such as deblurring [55], video frame synthesis [38] or super-resolution [29, 52], but their application to video denoising has been limited so far. In [13] a recurrent architecture is proposed, but the results are below the state of the art. More recently, Tassano *et al.* [58] proposed DVDnet, a convolutional architecture which processes five consecutive frames to predict the central frame. Each frame is first denoised spatially, and then warped to frame t via an optical flow. The aligned frames are stacked together with the central frames and processed by a "temporal denoising" network. The authors use a non-trainable optical flow, which prevents the network from being trained end-to-end. Nevertheless, this network produces state-of-the-art results. Two recent works proposed networks without explicit motion estimation: ViDeNN-G [15] processes three consecutive frames, and applies first a spatial denoising followed by temporal denoising, similar to [58], except that the frames are stacked without aligning. A different architecture, named fastDVDnet, was proposed in [59]. Instead of first using a spatial denoising, three consecutive noisy frames are stacked together (*early fusion*). The stack is processed by a U-net [49] which predicts the central frame. In order to extend the temporal receptive field of the network, the authors cascade two levels of these networks. The overall network takes five frames as input. Some works have tackled the related problem of burst denoising. Recently [23, 27, 43] focused on the related problem of image burst denoising reporting very good results. There is also recent work focusing on unknown noise-model denoising in a totally blind fashion [24] with videos.

In addition to CNNs, patch-based methods also yield state-of-the-art results [3, 9, 17, 22, 40, 61]. They exploit extensively the self-similarity of natural images and videos, namely the fact that most patches have several similar patches around them (spatially and temporally). Each patch is denoised using these similar

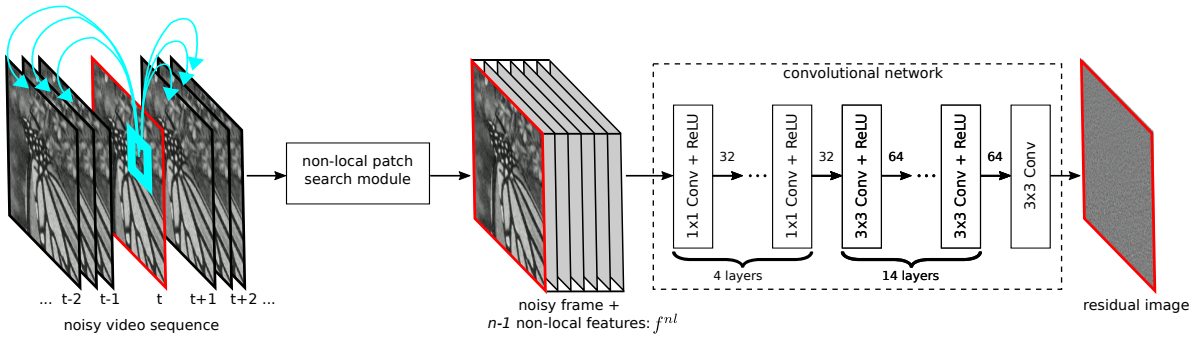

Fig. 1: Illustration of the proposed *Video Non-Local Network* (VNLnet). The first module performs a patch-wise nearest neighbor search across neighboring frames. Then, the current frame, and the feature vectors f^{nl} of each pixel (the center pixels of the nearest neighbors) are fed into the network. The first four layers of the network perform 1×1 convolutions with 32 feature maps. The resulting feature maps are the input of a simplified DnCNN [64] network with 15 layers (the number of features differs for RGB videos).

patches, which are searched for in a region around it. The search region generally is a space-time cube, but more sophisticated search strategies using optical flow have also been used. Because of the use of such broad search neighborhoods these methods are called *non-local*. While these video denoising algorithms perform very well, they often are computationally costly. Because of their complexity they are arguably unfit for high resolution video processing.

Patch-based methods usually follow three steps that can be iterated: (1) search for similar patches, (2) denoise the group of similar patches, (3) aggregate the denoised patches to form the denoised frame. VBM3D [17] improves the image denoising algorithm BM3D [18] by searching for similar patches in neighboring frames using a "predictive search" strategy which speeds up the search and gives some temporal consistency. VBM4D [40] generalizes this idea to 3D patches. In VNLB [2], video extension of [34], spatio-temporal patches that were not motion compensated are used to improve the temporal consistency. In [22] a generic search method extends every patch-based denoising algorithm into a global video denoising algorithm by extending the patch search to the entire video. SPTWO [9] and DDVD [8] use optical flow to warp the neighboring frames to each target frame. Each patch of the target frame is then denoised using the similar patches in this volume with a Bayesian strategy similar to [34]. Recently, [61] proposed to learn an adaptive optimal transform using batches of frames.

Patch-based approaches also achieve the state of the art among frame-recursive methods [4, 25]. These methods compute the current frame using only the current noisy frame and the previous denoised frame.

Contributions. In this work we propose a strategy to exploit non-local information with a CNN in the context of image and video denoising. It works particularly well in the case of video denoising, where it achieves state-of-the-art results.

The method first computes for each image patch the n most similar neighbors in a rectangular spatio-temporal search window and gathers the center pixel of each similar patch forming a feature vector which is assigned to each image location. This results in an image with n channels, which is fed to a CNN trained to predict the clean image from this high dimensional vector. We trained our network for grayscale and color video denoising. Practically training this architecture is made possible by a GPU implementation of the patch search that allows computing the nearest neighbors efficiently. The temporal self-similarity present in videos enables strong denoising results with our proposal.

To train our network we created a dataset of 17k video segments. In the two testing datasets, our network obtains state-of-the-art results on both color and grayscale video denoising. The code to generate the datasets and reproduce our results is available online[1]. A preliminary version of this work was presented in [20]. The present version includes an extension to color videos, a detailed comparison with recent state-of-the-art works, extended discussions comparing these methods, and new experiments.

2 Proposed method

Let u be a video and $u(x,t)$ denote its value at pixel position x in frame t. We observe v, a noisy version of

[1] The code to reproduce our results, the training and testing datasets can be found at `https://github.com/axeldavy/vnlnet`.

u contaminated by additive white Gaussian noise:

$$v = u + r, \quad \text{where, } r(x,t) \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

Our video denoising network processes the video frame by frame. Before it is fed to the network, each frame is pre-processed by a non-local patch search module which computes a non-local feature vector at each image position. The resulting non-local feature vector is then fed to the network. A diagram of the proposed method is shown in Figure 1. We call our method VNLnet for *Video Non-Local Network*.

2.1 Non-local features

Each frame is pre-processed by the non-local patch search module to produce a 3D tensor f^{nl} of n channels. This is the input to the network. The parameter n is the number of nearest neighbor patches searched for each pixel by this pre-processing module.

Let P be a patch centered at pixel x in frame t. The patch search module computes the distances between the patch P and the patches in a 3D rectangular search region of size $w_s \times w_s \times w_t$ centered at (x,t), where w_s and w_t are the spatial and temporal sizes. The positions of the n most similar patches are (x_i, t_i), ordered either by increasing distance or by increasing frame index t_i (see Section 4.2).

The pixel values at those positions are gathered to produce the n-dimensional non-local feature vector associated to pixel (x,t):

$$f^{\text{nl}}(x,t) = [v(x_1,t_1), ..., v(x_n,t_n)]. \tag{2}$$

Note that one of the channels of the feature images corresponds to the noisy image v, as the reference patch P is always among the n nearest neighbors.

2.2 Network architecture

For processing the non-local features, we use a network that is conceptually divided in two stages: a non-local stage and a local stage. The non-local stage consists of four 1×1 convolution layers with 32 kernels. The rationale for these layers is to allow the network to compute pixel-wise features out of the raw non-local features f^{nl} at the input.

The second stage receives the features computed by the first stage. It consists of 14 layers with 64 3×3 convolution kernels, each one followed by batch normalization and ReLU activations. The output layer is a 3×3 convolution. Its architecture is similar to the DnCNN network introduced in [64], although with 15 layers instead of 17 (as in [65]). As for DnCNN, the network outputs a residual image, which has to be subtracted to the noisy image to get the denoised one. For RGB videos, we use the same number of layers, but triple the number of features for each layer.

The training loss is the mean square error (MSE) between the reconstructed frame and the ground truth. We denote by $\mathcal{F}_\theta$ the application of the network. The input to $\mathcal{F}_\theta$ at time t is $f^{\text{nl}}_t = f^{\text{nl}}(\cdot, t)$, the n-channel image with non-local features gathered from a window of frames around t. The denoised frame is obtained by subtracting the noise predicted by the network from the noisy frame $v_t = v(\cdot, t)$, i.e. $\hat{u}_t = v_t - \mathcal{F}_\theta(f^{\text{nl}}_t)$. Then the training loss l is

$$l(\mathcal{F}_\theta(f^{\text{nl}}_t), u_t) = ||v_t - \mathcal{F}_\theta(f^{\text{nl}}_t) - u_t||_2^2. \tag{3}$$

Equivalently, this loss can be seen as the MSE between the predicted residual $\mathcal{F}_\theta(f^{\text{nl}}_t)$ and the added noise $v_t - u_t$.

The proposed non-local features could be used in conjunction with other network architectures proposed for image denoising. This would require of course changing the input layer, and retraining the network for the new input.

3 Datasets and training

3.1 Datasets

For the training and validation sets we used a database of short segments of 16 frames extracted from YouTube videos. Only HD videos with Creative Commons license were used. From each video we extracted several segments, separated by at least $10s$. In total the database consists of 16950 segments extracted from 1068 videos, organized in 64 categories (such as antelope, cars, factory, etc.). As the original videos might contain compression artifacts, noise, etc, we downscaled the video (to a height of 540 pixels). This removes the minor artifacts of the videos and better represents clean targets. In addition, we randomized the anti-aliasing filter width (Gaussian blur) of the downscaling. This results in a variety of sharpness/blur in the training dataset, and thus helps reducing dataset bias. We separated 6% of the videos of the database for the validation (one video for each category). For grayscale networks, grayscale data is obtained by converting the previous color datasets.

For training we ignored the first and last frames of each segment for which the 3D patch search window did not fit in the video. During validation we only considered the central frame of each sequence. The resulting validation score is thus computed on 503 sequences (1 frame each).

For testing, we used two distinct datasets. The first one is the dataset of [1], which was extracted from the Derf's Test Media collection.[2] It is composed of seven sequences of 100 frames of size 960×540. In this dataset, the camera is either still or has a smooth motion. The second one is the `test-dev` split of the DAVIS video segmentation challenge [45]. It consists of 30 videos having between 25 and 90 frames. In this dataset, the motion is more challenging. In order to remove compression artifacts and noise present in the original images, both datasets were obtained with a similar downscaling as for the training set (the original images ranged between HD and 4K). Each dataset was processed using a different anti-aliasing filter width.

3.2 Training

At each training epoch, first a subset of the videos of the dataset is selected and noise is added to generate noisy samples. Second the non-local patch search module is run on every video selected. This results in *videos of non-local features* where each frame has n channels containing the output of the patch search module. Third the network is trained on mini-batches built from small crops extracted at random positions on the videos of non-local features.

During training, we ignore spatio-temporal border effects by excluding the first and last $w_t/2$ frames and ignoring crops at borders. At testing time, we simply extended the video by mirroring it at the start and the end of the sequence and adding black borders for the patch-search module.

An epoch comprised 14000 batches of size 128, composed of square crops of 44 pixels width. We trained for 20 epochs with Adam [32] and reduced the learning rate at epochs 12 and 17 (from $1e^{-3}$ to $1e^{-4}$ and $1e^{-6}$ respectively). Training a network took 16 hours on a NVIDIA TITAN V for grayscale videos, and 72 hours for color videos.

4 Experiments and parameter tuning

In this section we evaluate the effect of the non-local features first in still image denoising, and then, after studying the impact of the parameters, in video denoising.

[2] `https://media.xiph.org/video/derf`

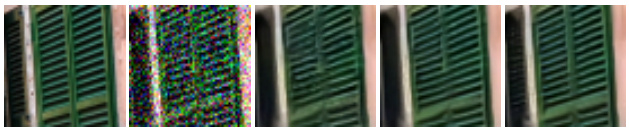

Fig. 2: Results on still image denoising (AGWN with $\sigma = 25$). Original clean image, noisy image, result obtained with the baseline CNN, result of incorporating the non-local features by finding the nearest neighbors on the noisy image or the oracle noise-less image. Contrast has been linearly scaled for better visualization.

4.1 The potential of non-locality

Although the focus of this work is in video denoising, it is still interesting to study the performance of the proposed non-local CNN on images. Figure 2 shows a comparison of a baseline CNN (a 15 layer version of DnCNN [64], as in our network) and a version of our method trained for still image denoising (it collects 9 neighbors by comparing 9×9 patches). The non-local features are sorted by patch distance. The results with and without non-local information are very similar. The only differences are visible on very self-similar parts like the shutters in Figure 2. This is confirmed by quantitative results. The average PSNR on the CBSD68 dataset [42, 64] (noise with $\sigma = 25$) obtained for the baseline CNN is of 31.24dB. The non-local CNN only leads to a 0.04dB improvement (31.28dB). The figure also shows the result of an oracular method: the nearest neighbor search is performed on the noise-free image, though the pixel values are taken from the noisy image. The method obtains an average PSNR of 31.85dB, 0.6dB over the baseline. The oracular results show that non-locality has a great potential to improve the results of CNNs. However, this improvement is hindered by the difficulty of finding accurate matches in the presence of noise. A way to reduce the matching errors is to use larger patches. But on images, larger patches have fewer similar patches. In contrast, as we will see below, the temporal redundancy of videos allows using very large patches.

4.2 Parameter tuning for video denoising

The non-local search has three main parameters: The patch size, the number of retained matches and the number of frames in the search region. We expect the best matches to be past or future versions of the current patch, so we set the number of matches as the number of frames on which we search. The non-local features are ordered based on patch distance.

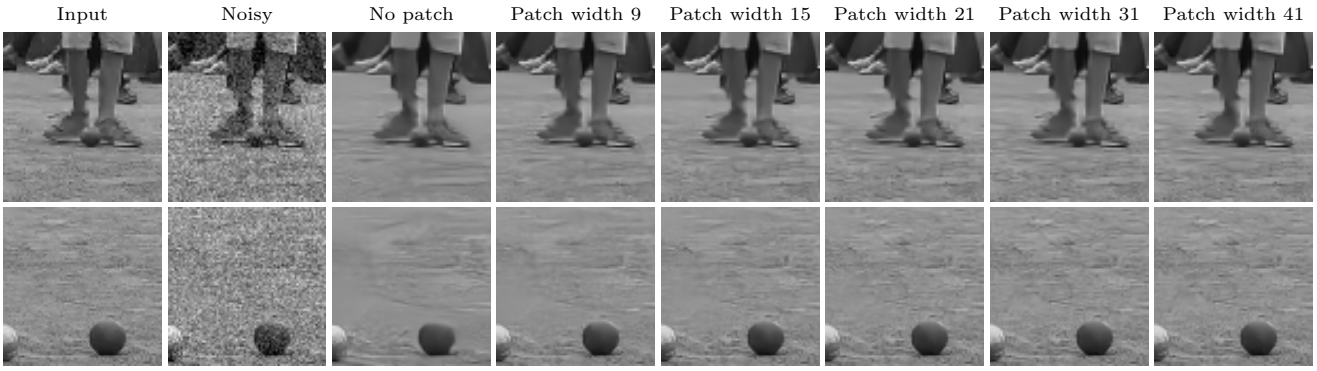

Fig. 3: Example of denoised results with our method when changing the patch size, respectively no patch search, 9×9, 15×15, 21×21, 31×31 and 41×41 patches. The 3D search window has 15 frames for these experiments.

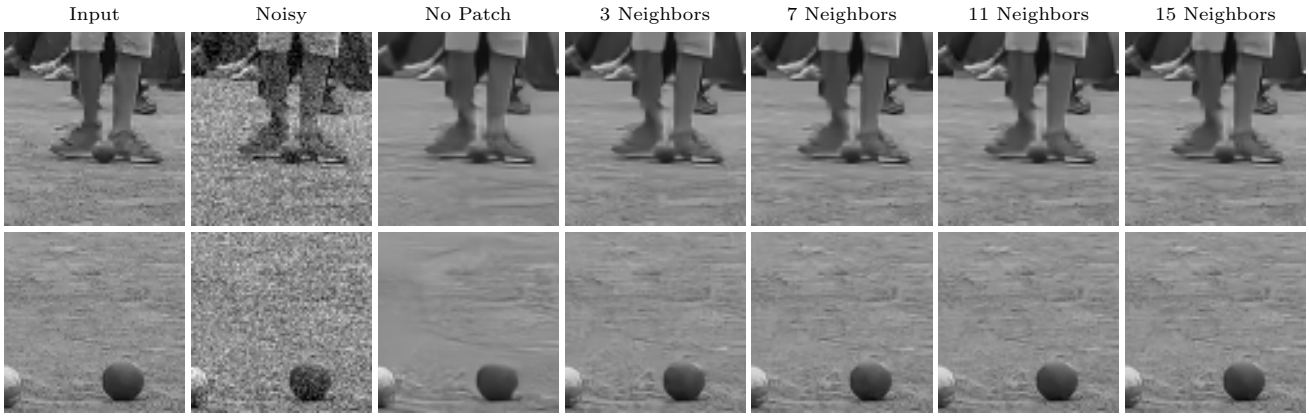

Fig. 4: Example of denoised results with our method when changing the number of frames considered in the 3D search window (respectively no patch search, 3, 7, 11 and 15). 41×41 patches were used for these experiments.

Patch size	no patch	9×9	15×15	21×21	31×31	41×41
PSNR	33.75	35.62	36.40	36.84	37.11	37.22

(a) Impact of the patch size (with 15 frames)

# search frames	no patch	3	7	11	15
PSNR	33.75	35.35	36.50	36.97	37.22

(b) Impact of number of frames (patch size is 41×41)

Patch search	no restriction	one neighbor per frame
PSNR	37.22	37.46

(c) Enforcing one neighbor per-frame

Table 1: Effect of patch size and number of frames (the number of neighbors is kept equal to the number of frames). In Table 1c we use 15 neighbors (and frames) and a patch size 41×41. PSNR computed on the validation set for noise with $\sigma = 20$.

4.2.1 Patch size

In Table 1a, we explore the impact of the patch size used for the matching. Figure 3 shows corresponding visual results. Surprisingly, we obtain better and better results by increasing the size of the patches. The main reason for this is that the match precision is improved, as the impact of noise on the patch distance shrinks. However, the disadvantage of using big patches is that the patch search can be affected by the motion of surrounding regions and pick a different motion to the one of the center pixel. Fortunately, the network seems to learn to identifying when the patch search fails and revert to a single-frame denoising like DnCNN.

This phenomenon can be observed in Figure 5, where we compare the results of the proposed VNLnet against the single frame DnCNN from [64] and two other meth-

# stacked frames	1	3	7	11	15
PSNR	33.75	35.54	36.30	36.54	36.56

Table 2: Impact of the number of frames considered for the *video-DnCNN* network (the network input is a 3D crop rather than the result of the non-local search). PSNR on the validation set AWGN with $\sigma = 20$.

ods: The *Non-Local Pixel Mean*, which simply averages the values of the non-local features given by the patch-matching, and *video-DnCNN*, an extension to video of DnCNN that will be described in 4.3. On the first two rows, the motion is consistent on the whole crop, and thus the Non-Local Pixel Mean output is sharp, indicating good matches. As a result, VNLnet's output shows more details than the corresponding DnCNN output. However the Non-Local Pixel Mean output is blurry for the person in the third row and the background of the fourth row. For these regions, VNLnet's output has a similar quality over DnCNN. Meanwhile, for the regions of these two crops with good matches (the background of the third row, and the person of the fourth row), the quality is improved. Using bigger patches will increase the number of patches covering regions of conflicting motion. As a result, we see that the performance gain from 31×31 to 41×41 is rather small.

4.2.2 One neighbor per frame

With such large patches, only matches of the same objects from different frames are likely to be taken as neighbors. Thus we go a step further by enforcing matches to come from different frames. In this variant, neighbors are sorted by frame index instead of the patch distance. Note the network is retrained as the patch distribution is impacted. The resulting network produces a slight performance improvement, shown in Table 1c.

With this configuration, the ith non-local feature map corresponds to a warped version of the ith neighboring frame, aligned to the reference frame t. This can be related to [58, 62], which align the input frames using an optical flow. Indeed patch matching can be seen as a rough optical flow with integer displacements. A natural question would be if better results could be obtained using a standard optical flow. The results of Table 1a highlight the importance of very reliable matches, and thus the optical flow would have to be chosen with care. In [59], the authors of DVDnet [58] stressed the difficulty of finding a fast and reliable optical flow, and moved away from it for FastDVDnet [59]. In [62], the optical flow is computed with a reduced version of SpyNet [47], which is trained together with the rest of the network. In our case, the patch search module is not trainable, and the network is trained to process its output. The main challenge here is the convergence of the training to a good local minimum in the presence of noise and when considering a large number of neighboring frames. These questions will be addressed in future works.

4.2.3 Number of frames

In Table 1b and Figure 4, we see the impact of the number of frames used in the search window (and thus the number of nearest neighbors). One can see that the more frames, the better. Increasing the number of frames beyond 15 (7 past, current, and 7 future) does not justify the small increase of performance. Foreground moving objects are unlikely to get good neighbors for the selected patch size, unlike background objects, thus it comes to no surprise that the visual quality of the background improves with the number of patches, while foreground moving objects (for example the legs in Figure 4) do not improve much.

In the following experiments, we shall use 41×41 patches and 15 frames. Another parameter for non-local search is the spatial width of the search window, which we set to 41 pixels (the center pixel of the tested patches must reside inside this region). We trained grayscale and color networks for AGWN of σ 10, 20 and 40. To highlight the fact that a CNN method can adapt to many noise types, unlike traditional methods, we also trained a grayscale network for Gaussian noise correlated by a 3×3 box kernel such that the final standard deviation is $\sigma = 20$, and 25% uniform Salt and Pepper noise (removed pixels are replaced by random uniform noise).

4.3 A note on motion handling

To evaluate the effectiveness of the non-local features we also train a network with the same architecture, but instead of feeding it with the n non-local features we feed it with the stack of the n neighboring frames. We call this network video-DnCNN. We do this for different values of n, and show the results obtained on Table 2.

The performance of the video-DnCNN network increases with the number of frames, although less than the proposed VNLnet (see Table 1b). In particular, for video-DnCNN the average PSNR on the validation set stagnates at 11 frames. The reason for this stagnation is that while the denoising performance increases on sequences with majority of small and smooth motions, it drops significantly when there are many large or irregular motions.

Without the non-local patch-search module, the network has to learn to handle motion implicitly, which makes the task significantly harder. As the number of input frames increases, so does the complexity of the internal motion compensation the network has to learn to denoise accurately. The video-DnCNN network then overfits to the most frequent motion patterns in the

Fig. 5: Example of denoised result for *Non-Local Pixel Mean*, DnCNN, video-DnCNN (see Section 4.3) and VNLnet, for AGWN with $\sigma = 20$. The four crops highlight the results on frames feature various kinds of motion. The videos are part of the DAVIS dataset. *Non-Local Pixel Mean* corresponds to the average of the output of the non-local patch search.

training set, and fails when it encounters a different motion.

This can be seen in Figure 5, by comparing the results of video-DnCNN and VNLnet, for objects with fast/irregular motion patterns. VNLnet is able to recover much more details, thanks to the patch search.

Figure 6 shows the PSNR gain of VNLnet over video-DnCNN for each sequence on the grayscale DAVIS test set. The gain given by the non-local patch-search module is significant, except only for two sequences. These feature fixed cameras and static backgrounds covering most of the frame. The sequences with larger gains have complex motion. By factoring out the motion, the non-local patch search module removes the need for the network to learn to adapt to various types of motion, enabling a better generalization on various moving scenes.

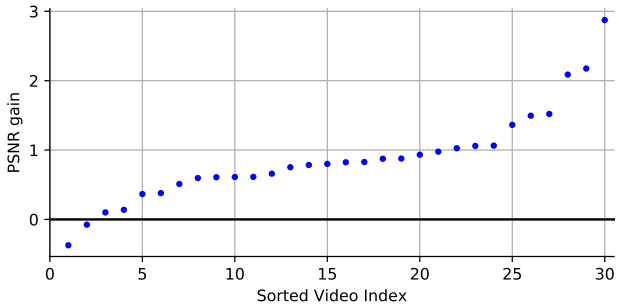

Fig. 6: PSNR gain of VNLnet versus the network without patch search of Table 2 (15 input frames) on grayscale DAVIS ($\sigma = 20$) (35.46db versus 34.58db).

5 Comparison with the state of the art

In this section, we compare the proposed method VNLnet to VBM3D [17], VNLB [2], and DnCNN [64] for

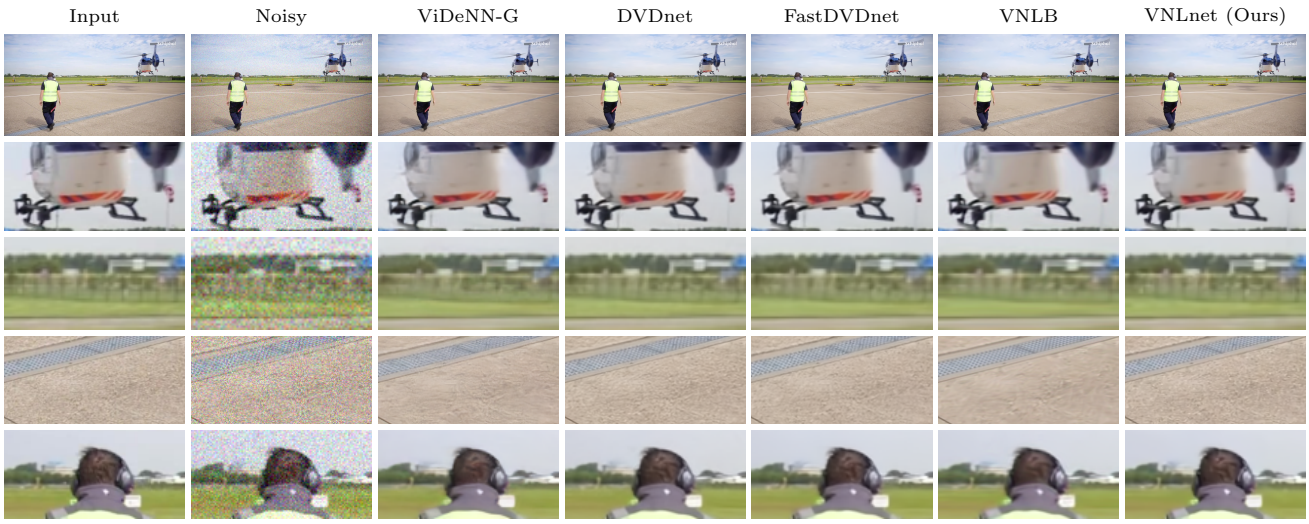

Fig. 7: Example of denoised result for several algorithms (AGWN with $\sigma = 20$) on a sequence of the color DAVIS dataset [45]. The crops highlight the results on non-moving and moving parts of the video.

		Method	$\sigma = 10$	$\sigma = 20$	$\sigma = 40$
GRAYSCALE	DERF	SPTWO	39.56 / .9675	35.99 / .9368	30.93 / .7901
		VBM3D	38.24 / .9599	34.68 / .9100	31.11 / .8360
		VBM4D	38.88 / .9534	35.10 / .9169	31.40 / .8432
		VNLB	**40.57** / ***.9731***	**36.81** / **.9428**	**32.95** / **.8856**
		DnCNN	37.28 / .9482	33.60 / .8973	30.09 / .8156
		VNLnet	***40.21*** / **.9732**	***36.47*** / ***.9414***	***32.51*** / ***.8752***
			Corr. Gaussian noise		Uniform S&P 25%
		VNLB	***25.39*** / ***.5922***		***23.49*** / ***.7264***
		VNLnet	**30.94** / **.9452**		**48.12** / **.9951**
	DAVIS		$\sigma = 10$	$\sigma = 20$	$\sigma = 40$
		VBM3D	37.43 / .9425	33.75 / .8870	30.12 / .8068
		VNLB	***38.84*** / ***.9634***	***35.26*** / ***.9240***	***31.88*** / ***.8622***
		DnCNN	36.80 / .9451	32.94 / .8878	28.69 / .7940
		VNLnet	**39.07** / **.9663**	**35.46** / **.9299**	**31.90** / **.8659**
COLOR	DERF	VBM3D	38.19 / .9560	34.80 / .9165	31.65 / .8568
		VNLB	**40.93** / **.9760**	**37.62** / ***.9528***	***33.97*** / .9042
		DnCNN	38.00 / .9588	34.44 / .9171	31.14 / .8520
		ViDeNN-G	38.16 / .9588	35.34 / .9291	32.25 / .8757
		DVDnet	39.08 / .9689	36.48 / .9474	33.43 / ***.9051***
		FastDVDnet	39.01 / .9669	36.16 / .9427	33.21 / .9010
		VNLnet	***40.49*** / ***.9749***	***37.46*** / **.9549**	**34.02** / **.9117**
	DAVIS	VBM3D	38.43 / .9591	34.74 / .9157	31.38 / .8473
		VNLB	***40.31*** / ***.9725***	***36.79*** / .9420	33.34 / .8896
		DnCNN	38.91 / .9655	35.24 / .9278	31.81 / .8637
		ViDeNN-G	38.46 / .9619	35.47 / .9314	32.32 / .8756
		DVDnet	39.31 / .9702	36.66 / ***.9488***	***33.61*** / ***.9059***
		FastDVDnet	39.74 / .9714	36.50 / .9457	33.35 / .9013
		VNLnet	**40.71** / **.9760**	**37.39** / **.9534**	**33.96** / **.9091**

Table 3: Quantitative comparison (PSNR and SSIM) of state-of-the-art methods versus the proposed VNLnet on two test sets, both in grayscale and in color. We highlighted the best performance in bold and the second best in bold italics.

grayscale videos, and VBM3D [17], VNLB [2], DnCNN [64], ViDeNN-G [15], DVDnet [58], and FastDVDnet [59] for color videos. DnCNN was applied frame-by-frame.

Table 3 shows the denoising results obtained on the two compared datasets. For grayscale videos, we also include results for SPTWO [9] and VBM4D [39], computed in [1] for the DERF dataset. Figures 7 and 8 show results for the most relevant color methods. VNLB (Video Non-Local Bayes) outperformed on average all other methods on the DERF dataset, except at noise $\sigma = 40$ in color where VNLnet performed better. Meanwhile on the DAVIS dataset, our method performed the best both in grayscale and color, for all the three tested noise levels. VNLB ranked second, except in color for high noise levels, where it was surpassed by DVDnet and FastDVDnet.

A comparison of grayscale and color results in Table 3 reveals that CNN-based methods exploit better the correlations between color channels: while for grayscale, VBM3D significantly outperforms DnCNN in PSNR on the DAVIS dataset, the reverse occurs for color. Moreover, VNLnet performed proportionally better in color than in grayscale. This should not come as a surprise, since the way in which VBM3D and VNLB treat color is rather heuristic: an orthogonal color transform is applied to the video which is supposed to decorrelate color information. Then the processing of each color channel of a group of patches is done independently.

In order to better compare qualitative aspects of the results we show some details in Figures 7 and 8. For some scenes, VNLnet recovers significantly more details in the background, as shown in Figure 7. In general, we observe that VNLnet, and the other video CNN methods (ViDeNN-G, DVDnet, and FastDVDnet) have better background reconstruction than VNLB. This can be seen in Figure 7 and Figure 8. Some details however are better recovered by VNLB and VNLnet. For example in Figure 8 both methods recover the red lights in the

Fig. 8: Examples of areas where the level of restored detail of the methods differs significantly (AGWN with $\sigma = 40$) in videos of DERF.

top left corner of the image in the first column, while for the three other methods the lights do not appear. In the second column, VNLB does not reconstruct the trees as well as the CNN-based methods, but manages to recover the color of the clothes of the person in the bottom left. VNLnet not only recovers better the tree structure, but also recovers the clothes correctly. None of the methods restore satisfyingly the grass texture in the third column of Figure 8 for the tested noise level. This highlights that there is room for improvement. All five methods achieve reasonable temporal consistency, which is an important quality requirement for video denoising.

One of the benefits of CNNs over traditional model-based approaches is that they can be easily retargeted to handle other noise models. To illustrate this we compare VNLB and our method on non-standard noises in Table 3. As expected, VNLnet significantly outperforms VNLB for these non-Gaussian noise distributions.

In summary, the proposed approach for video denoising obtains state-of-the-art results on both test sets. In particular, it outperforms previous CNN approaches. In light of this, we can conclude that effectively exploiting a large number of surrounding frames is key. Indeed, the proposed VNLnet uses 15 frames, in comparison to the 3 frames used by ViDeNN-G and 5 frames of DVDnet and FastDVDnet. Most of these methods avoid relying on an explicit optical flow computation, which can be unreliable given that the input frames are noisy. FastDVDnet and ViDeNN-G do so by performing an early fusion of triplets of consecutive frames without alignment. The non-local features computed via patch correspondences result in an effective way to present the information of large frame neighborhoods to a network that merges them. Training the merging network is then straightforward.

5.1 Running times

In Tables 4 and 5, we compare the running time in grayscale and color of several methods when denoising a video frame. In Table 4, the compared methods are run on a single CPU core, while in Table 5, a system with an Intel Xeon W-2145 and a NVIDIA TITAN V is used. For both tables, the loading and writing time of the videos were subtracted. In addition, while we do not a have a CPU implementation of the patch search layer, from the GPU runtimes of Table 6 we can expect that on CPU our method should be 10 times slower than DnCNN. The non-local search is particularly costly because matches are searched in 15 frames for patches centered at every pixel of our image. The patch search implemented is similar to the convolution-based patch search described in [19]. The implementation could be accelerated by reducing the size of the 3D window using tricks explored in other papers. VBM3D for example centers the search on each frame on small windows around the best matches found in the previous frame. A related acceleration is to use a search strategy based on PatchMatch [6].

VBM3D	DnCNN	VBM4D	VNLB	SPTWO
1.3s	13s	52s	140s	210s

Table 4: Running time per frame on a grayscale 960 × 540 video for VBM3D, DnCNN, VBM4D, VNLB and SPTWO on a single CPU core.

VNLB	ViDeNN-G	DVDnet	FastDVDnet	VNLnet
26.94s	0.186s	2.67s	0.074s	1.42s

Table 5: Running time per frame on a color 960 × 540 video for VNLB, ViDeNN, DVDnet, FastDVDnet, VNLnet on a system with a 16-cores CPU (Intel Xeon W-2145) and a NVIDIA TITAN V.

Non-local search	Rest of the network	DnCNN
850 ms	80 ms	95 ms

Table 6: Running time per frame on a grayscale 960 × 540 video on a NVIDIA TITAN V (41 × 41 patches at every position, 41 × 41 × 15 3D windows, the default parameters).

6 Conclusions

We described an effective way of incorporating temporal non-local information into a CNN for video denoising. The proposed method computes for each image patch the n most similar neighbors on a spatio-temporal window and gathers the value of the central pixel of each similar patch to form a non-local feature vector which is given to a CNN. Our method yields a significant gain compared to other CNN approaches. It has similar performance to the best non-CNN method evaluated, VNLB, outperforming it on the largest of our test datasets. In addition, we noted that CNN approaches tend to better reconstruct backgrounds than VNLB, which are perceptually relevant areas. To prevent dataset bias we also proposed a public training set comprising 17k videos from 64 different categories and a simulation strategy that emulates different levels of sharpness.

Our contribution places neural networks among the best video denoising methods and opens the way for

new works in this area. In particular we have seen the importance of having reliable matches: On the validation set, the best performing method used patches of size 41×41 for the patch search. We have also noticed that on regions with non-reliable matches (complex motion), the network seems to revert to a result similar to single image denoising. Thus we believe future works should focus on improving this area.

References

1. Arias, P., Facciolo, G., Morel, J.M.: A comparison of patch-based models in video denoising. In: IVMSP. IEEE (2018)
2. Arias, P., Morel, J.M.: Towards a bayesian video denoising method. In: ACIVS, LNCS. Springer (2015)
3. Arias, P., Morel, J.M.: Video denoising via empirical bayesian estimation of space-time patches. JMIV **60**(1), 70–93 (2018). DOI 10.1007/s10851-017-0742-4. URL `https://doi.org/10.1007/s10851-017-0742-4`
4. Arias, P., Morel, J.M.: Kalman filtering of patches for frame-recursive video denoising. In: IEEE CVPRW (2019)
5. Barbu, A.: Training an active random field for real-time image denoising. IEEE TIP **18**(11), 2451–2462 (2009). DOI 10.1109/TIP.2009.2028254
6. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch. In: SIGGRAPH. ACM Press (2009). DOI 10.1145/1576246.1531330. URL `http://dx.doi.org/10.1145/1576246.1531330http://portal.acm.org/citation.cfm?doid=1576246.1531330`
7. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. Computer Vision and Pattern **2**, 60–65 (2005). DOI 10.1109/CVPR.2005.38
8. Buades, A., Lisani, J.L.: Dual domain video denoising with optical flow estimation. In: IEEE ICIP (2017). DOI 10.1109/ICIP.2017.8296830
9. Buades, A., Lisani, J.L., Miladinović, M.: Patch-based video denoising with optical flow estimation. IEEE TIP **25**(6), 2573–2586 (2016). DOI 10.1109/TIP.2016.2551639
10. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: IEEE CVPR (2012). DOI 10.1109/CVPR.2012.6247952
11. Chambolle, A., Caselles, V., Cremers, D., Novaga, M., Pock, T.: An introduction to total variation for image analysis. Theoretical foundations and numerical methods for sparse recovery **9**(263-340), 227 (2010)
12. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: IEEE CVPR (2018)
13. Chen, X., Song, L., Yang, X.: Deep rnns for video denoising. In: Applications of Digital Image Processing (2016)
14. Chen, Y., Pock, T.: Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. IEEE PAMI **39**(6), 1256–1272 (2017)
15. Claus, M., van Gemert, J.: Videnn: Deep blind video denoising. In: IEEE CVPRW (2019)
16. Cruz, C., Foi, A., Katkovnik, V., Egiazarian, K.: Nonlocality-reinforced convolutional neural networks for image denoising. IEEE SPL **25**(8), 1216–1220 (2018). DOI 10.1109/LSP.2018.2850222
17. Dabov, K., Foi, A., Egiazarian, K.: Video denoising by sparse 3D transform-domain collaborative filtering. In: EUSIPCO (2007)
18. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE TIP (2007)
19. Davy, A., Ehret, T.: Gpu acceleration of nl-means, bm3d and vbm3d. Journal of Real-Time Image Processing pp. 1–18 (2020)
20. Davy, A., Ehret, T., Morel, J.M., Arias, P., Facciolo, G.: A non-local cnn for video denoising. In: IEEE ICIP (2019)
21. Donoho, D.L., Johnstone, J.M.: Ideal spatial adaptation by wavelet shrinkage. Biometrika **81**(3), 425–455 (1994). DOI 10.1093/biomet/81.3.425. URL `http://biomet.oxfordjournals.org/content/81/3/425.abstract`
22. Ehret, T., Arias, P., Morel, J.M.: Global patch search boosts video denoising. In: VISAPP (2017)
23. Ehret, T., Davy, A., Arias, P., Facciolo, G.: Joint demosaicing and denoising by overfitting of bursts of raw images. In: IEEE ICCV (2019)
24. Ehret, T., Davy, A., Morel, J.M., Facciolo, G., Arias, P.: Model-blind video denoising via frame-to-frame training. In: IEEE CVPR (2019)
25. Ehret, T., Morel, J., Arias, P.: Non-Local Kalman: A recursive video denoising algorithm. In: IEEE ICIP (2018)
26. Facciolo, G., Pierazzo, N., Morel, J.: Conservative scale recomposition for multiscale denoising (the devil is in the high frequency detail). SIIMS **10**(3), 1603–1626 (2017). DOI 10.1137/17M1111826
27. Godard, C., Matzen, K., Uyttendaele, M.: Deep burst denoising. In: ECCV (2018)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE CVPR (2016)

29. Huang, Y., Wang, W., Wang, L.: Bidirectional recurrent convolutional networks for multi-frame super-resolution. In: NIPS (2015)
30. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: ICML (2015)
31. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In: NIPS (2009)
32. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
33. Kobler, E., Klatzer, T., Hammernik, K., Pock, T.: Variational networks: Connecting variational methods and deep learning. In: Pattern Recognition, pp. 281–293. Springer (2017)
34. Lebrun, M., Buades, A., Morel, J.M.: A nonlocal bayesian image denoising algorithm. SIIMS (2013)
35. Lebrun, M., Colom, M., Buades, A., Morel, J.M.: Secrets of image denoising cuisine. Acta Numerica **21**(1), 475–576 (2012)
36. Lefkimmiatis, S.: Non-local color image denoising with convolutional neural networks. In: IEEE CVPR (2017)
37. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.S.: Non-local recurrent network for image restoration. In: NIPS (2018)
38. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: IEEE ICCV (2017)
39. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video Denoising Using Separable 4D Nonlocal Spatiotemporal Transforms. In: Proc. of SPIE (2011)
40. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. IEEE TIP (2012)
41. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: NIPS (2016)
42. 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: ICCV. IEEE (2001)
43. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. In: CVPR (2018)
44. Plötz, T., Roth, S.: Neural nearest neighbors networks. In: NIPS (2018)
45. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv:1704.00675 (2017)
46. Qiao, P., Dou, Y., Feng, W., Li, R., Chen, Y.: Learning non-local image diffusion for image denoising. In: ACMMM (2017)
47. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: IEEE CVPR (2017)
48. Romano, Y., Elad, M., Milanfar, P.: The little engine that could: Regularization by denoising (red). SIIMS **10**(4), 1804–1844 (2017). DOI 10.1137/16M1102884
49. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. Miccai pp. 234–241 (2015). DOI 10.1007/978-3-319-24574-4{_}28
50. Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In: IEEE CVPR (2005)
51. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D **60**(1-4), 259–268 (1992). DOI 10.1016/0167-2789(92)90242-F. URL `https://doi.org/10.1016/0167-2789(92)90242-F`
52. Sajjadi, M.S.M., Vemulapalli, R., Brown, M.: Frame-Recurrent Video Super-Resolution. In: IEEE CVPR (2018)
53. Santhanam, V., Morariu, V.I., Davis, L.S.: Generalized deep image to image regression. In: IEEE CVPR (2017)
54. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: IEEE CVPR (2014)
55. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: IEEE CVPR (2017)
56. Sun, J., Tappen, M.F.: Learning non-local range markov random field for image restoration. In: IEEE CVPR (2011)
57. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. IEEE ICCV (2017)
58. Tassano, M., Delon, J., Veit, T.: Dvdnet: A fast network for deep video denoising. In: IEEE ICIP (2019)
59. Tassano, M., Delon, J., Veit, T.: Fastdvdnet: Towards real-time video denoising without explicit motion estimation (2019). URL `http://arxiv.org/abs/1907.01361`
60. Vemulapalli, R., Tuzel, O., Liu, M.: Deep gaussian conditional random field network: A model-based deep network for discriminative denoising. In: IEEE CVPR (2016)
61. Wen, B., Li, Y., Pfister, L., Bresler, Y.: Joint adaptive sparsity and low-rankness on the fly: an online tensor reconstruction scheme for video denoising.

In: IEEE ICCV (2017)
62. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. IJCV **127**(8), 1106–1125 (2019). DOI 10.1007/s11263-018-01144-2. URL https://doi.org/10.1007/s11263-018-01144-2
63. Yang, D., Sun, J.: Bm3d-net: A convolutional neural network for transform-domain collaborative filtering. IEEE SPL **25**(1), 55–59 (2018). DOI 10.1109/LSP.2017.2768660
64. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE TIP **26**(7), 3142–3155 (2017). DOI 10.1109/TIP.2017.2662206. URL http://arxiv.org/abs/1608.03981http://ieeexplore.ieee.org/document/7839189/
65. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE TIP **27**(9), 4608–4622 (2018)
66. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: IEEE ICCV (2011)