Self-supervised training for blind multi-frame video denoising

Valéry Dewil   Jérémy Anger   Axel Davy   Thibaud Ehret   Gabriele Facciolo   Pablo Arias
Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, 91190, Gif-sur-Yvette, France
https://cmla.github.io/mf2f/

Abstract

We propose a self-supervised approach for training multi-frame video denoising networks. These networks predict each frame from a stack of frames around it. Our self-supervised approach benefits from the temporal consistency in the video by minimizing a loss that penalizes the difference between the predicted frame and a neighboring one, after aligning them using an optical flow. We use the proposed strategy to denoise a video contaminated with an unknown noise type, by fine-tuning a pre-trained denoising network on the noisy video. The proposed fine-tuning reaches and sometimes surpasses the performance of state-of-the-art networks trained with supervision. We demonstrate this by showing extensive results on video blind denoising of different synthetic and real noises. In addition, the proposed fine-tuning can be applied to any parameter that controls the denoising performance of the network. We show how this can be expoited to perform joint denoising and noise level estimation for heteroscedastic noise.

1. Introduction

Denoising has been a fundamental problem of image and video processing since the early days of these disciplines. It continues to be an active research area due to the continuous need for reducing the size of imaging sensors and the desire of imaging in increasingly challenging conditions (such as low light and short exposure times).

The current state of the art in image and video denoising is dominated by convolutional neural networks (CNNs) [64, 51, 35, 15, 55]. In addition to their superior performance, CNNs offer a greater flexibility as they can be trained to denoise potentially any type of noise [8, 58, 23, 6]. In contrast, traditional model-based approaches typically require a tractable model of the noise, and specific algorithms for each type of noise (*e.g.* [32, 19, 36, 49, 12, 3, 65]). This flexibility however, comes at a price, as it has been observed that CNNs are very sensitive to mismatches between the data and noise distributions at training and testing [43]. This has fueled the interest in training CNNs

for real noisy raw images, with the publication of several datasets and benchmarks [42, 43, 8, 1, 5], as well as methods [20, 4, 44, 27, 27, 63]. Most of this research focuses almost exclusively on still image denoising.

Producing datasets of realistic noisy-clean pairs for supervised training is a challenging task. Some works contaminate clean images with synthesized realistic noise [20, 4, 27, 63], but the results depend on the fit between the synthetic and real data. Realistic data can not be always be generated. For example the noise distribution might be unkown. Generative Adversarial Networks have been proposed to generate samples from a unknown noise distribution [9]. Other works have proposed datasets of real images with ground truth. For still image denoising, it is possible to acquire pairs of images of exactly the same scene, either altering the exposure time so that one of them is approximately noiseless [43, 8, 7], or by taking a second noisy shot with an independent noise realization as proposed by *noise-to-noise* [33]. Acquiring such pairs with real noise can be cumbersome and prone to dataset biases as the scenes need to be static. For video denoising the situation is even worse as it would require independent acquisitions of the exact same action [61].

A more ambitious goal is that of *self-supervised* training, where the network learns exclusively from noisy images/videos x_i with a loss that uses them both as input and target, for instance $\sum_i \|\mathcal{F}(x_i) - x_i\|^2$. To prevent the network from learning the identity function, restrictions are incorporated in the architecture. Denoising autoencoders [57] use a bottleneck forcing the network to filter out information. *Blind-spot* networks [28, 2] do not have access to the input pixel at j for computing the output pixel j (a blind spot at the center of the receptive field). This has a significant penalty on the performance, as the noisy value of a pixel is a valuable piece of information for denoising it. Some works re-introduce the blind spot in a second Bayesian estimation step [30, 29], but this requires knowing the noise distribution. A related approach is proposed in [46], where a fraction of input pixels is masked at random, and the network then learns to do joint denoising and inpainting. Averaging predictions obtained with different random masks

Figure 1: Denoising results on Poisson noise ($p = 8$), Box noise ($\sigma = 40, 3 \times 3$) and demosaicked Poisson noise ($p = 4$). From left to right: noisy frame; FastDVDnet [55] trained for each noise (supervised); DnCNN [64] fine-tuned with frame-to-frame (F2F) [18] (self-supervised); FastDVDnet fine-tuned with the offline version of the proposed multi-frame-to-frame (MF2F) framework (self-supervised).

leads to results comparable to supervised training. Blind-spot networks fail if the noise is spatially correlated. Spatially correlated noise is handled in [41], but it requires knowing the parameters (*e.g.* variance) of the noise distribution. Other works [38, 52] approximate the MSE risk with an unsupervised loss using Stein's unbiased risk estimator (SURE) [53]. Unfortunately, these approaches require the noise distribution to be known and cannot be applied to other risks.

The treatment of real noise in videos is beginning to attract more attention. Patch-based approaches have been proposed for handling correlated noise in compressed [34] and infrared videos [36]. In [39, 37, 11] CNNs are trained by synthesizing signal dependent noise. In [7] a still image denoising network is trained on low-light static sequences using a long exposure image as ground truth. The authors add a temporal consistency term to the loss to improve generalization to dynamic scenes. In [61] the authors train a video denoising network for raw video using a complex combination of simulated noise, datasets of long and short exposure raw images and a dataset of stop motion raw videos. In the latter each video is actually a sequence of static scenes. In this way several images can be captured for each video frame, and averaged to reduce the noise. Recently, Ehret et al. [18] proposed *frame-to-frame* (F2F), a method to fine-tune an image denoising network (or joint denoising and demosaicking [17]) to an unknown noise type from a single noisy video. The fine-tuning is based on a loss that penalizes the motion compensated error between the predicted frame and the previous noisy frame as *target*. The fine-tuned network achieved (and even surpassed) the performance of the same network trained with supervision for that specific noise. An important limitation of F2F is that its single-frame denoising network leads to sub-optimal video denoising results and lacks of temporal consistency.

Contributions. We introduce *multi-frame-to-frame* (MF2F), a self-supervised fine-tuning framework for video denoising networks that take a *stack* of several frames as input. The proposed fine-tuning allows to adapt a multi-frame network to an unknown noise type using a single noisy sequence. This extends the single-image F2F approach of [18] to multi-frame networks, resulting in a model blind video denoising method that achieves, *for the first time*, results on par with those of non-blind state-of-the-art methods.

Naively applying F2F to a multi-frame network leads to unwanted trivial solutions, as the target frame in the loss is part of the input stack. This applies to any loss in which the target is a function of the input to the network. We evaluate different configurations of non-overlapping input stacks and target frames and identify the ones yielding the best performance. We also found that the fine-tuned network leads to even better results if we switch back to the standard input
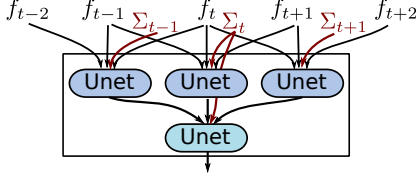
Figure 2: The FastDVDnet [55] architecture consists of two cascaded U-nets [47], each of which takes as input three frames (without alignment), plus a variance map Σ_i of the same size as the input frames. The first U-net is applied three times to produce initial estimates of the frames $t-1$, t and $t+1$. These estimates are then fed into the second network which predicts the central frame $\hat{u}_t$.

stack at inference time.

We call MF2F the fine-tuning method resulting from applying F2F with the proposed training stack. We demonstrate the effectiveness and flexibility of the proposed MF2F by fine-tuning a network pre-trained for additive white Gaussian noise (AWGN) to different noise types (AWGN, Poisson, colored Gaussian and demosaicked Poisson) and levels. The results are comparable to those of supervised noise-specific training (see Fig. 1). Evaluations on videos with real and realistic [4] camera noise show that MF2F outperforms state-of-the-art raw video denoising networks.

The proposed fine-tuning can also be applied to any network parameter which influences the denoising performance. We illustrate this by working with an AWGN denoising network that receives as input a noise variance map. Fine-tuning the variance map allows to jointly estimate the variance at each input pixel and denoise the video. We apply this to heteroscedastic Gaussian and Poisson noises. The proposed fine-tuning is able to recover complex variance maps with remarkable spatial resolution.

In Section 2 we describe the proposed framework. We validate our approach on synthetic data in Section 3 and on real noisy sequences in Section 4. Concluding remarks are given in Section 5.

2. Self-supervised Video Denoising

We consider a video f with frames f_t, that is a noisy version of a video u. The distribution of the noise is unknown. We assume that the noise at each frame is independent and median preserving in the noise-to-noise sense [33, 18].

Our self-supervised loss for video denoising extends the F2F loss introduced in [18], which penalizes the error between the output of the network at frame t with the noisy frame $t-1$ (the *target* frame). The authors of [18] consider a denoising network $\mathcal{F}_\theta$ which takes a single image as input, and train it via the following loss:

$$\ell_1^{\text{F2F}}\left(\mathcal{F}_\theta(f_t), f_{t-1}\right) = \|\kappa_t \circ (W_{t,t-1}\mathcal{F}_\theta(f_t) - f_{t-1})\|_1. \quad (1)$$

Here $\circ$ denotes the element-wise product, $W_{t,t-1}$ the warping operator from frame t to the target frame $t-1$, and κ_t
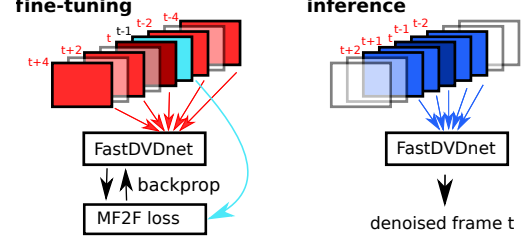

Figure 3: Proposed multiframe-to-frame blind fine-tuning for a video denoising network taking as input a stack of frames. During fine-tuning we use a dilated input stack (in red) so that the target frame is hidden from the network. At inference time we use the natural stack (in blue).

is an occlusion mask removing mismatches from the loss. Given the optical flow $v_{t-1,t}$, the warping operator from frame $t-1$ to t is defined as

$$(W_{t,t-1}u_t)(x) = u_t(x + v_{t-1,t}(x)), \quad (2)$$

where bicubic interpolation is used to resample the u_t.

In [18] the temporal consistency of the video is used to train the network but the network itself takes as input a single image. Better results can be obtained by network architectures that take into account temporal information. This can be done with frame recurrent networks [10] or by providing multiple frames as the input of the network [59, 14, 55]. We focus on the latter type of networks as they currently produce state-of-the-art results in video denoising. In particular, we will adopt the recent FastDVDnet [55] (see Fig. 2), but the proposed fine-tuning can be applied to other multi-frame networks as well. The FastDVDnet architecture is well suited as it takes five frames as input as well as variance maps. Furthermore, it can be trained end-to-end without requiring an external motion estimation stage.

We denote the input stack of frames as $\mathcal{S}_t = [f_{t-n}, ..., f_{t+n}]$. The tth denoised frame is produced as $\hat{u}_t = \mathcal{F}_\theta(\mathcal{S}_t)$ (to simplify the notation we will omit the input variance maps Σ_r with $r = t-1, t, t+1$).

2.1. Frame stacks for self-supervised training

The F2F loss (1) cannot be directly applied to $\mathcal{F}_\theta(\mathcal{S}_t)$, as it can be minimized simply by warping f_{t-1} (which is in the input stack) to $W_{t,t-1}^{-1}f_{t-1}$, *i.e.* by aligning the noisy frame f_{t-1} to f_t without removing the noise. In the following we show that any loss that depends only on the network input and its output leads to these unwanted solutions.

Suppose we want to predict y from z. In our case, z is a stack of noisy frames and y the clean version of the central frame of the stack, however the following arguments also apply to other regression problems. In a supervised training setting we minimize an approximation of the expected value of a loss penalizing the difference between the

Training stack $\mathcal{S}'_t$	ref.	Box	AWGN	Poiss.
$f_{t-2}, f_{t-1}, f_t, f_{t+1}, f_{t+2}$	f_{t-3}	28.93	28.78	28.13
$f_{t-3}, f_{t-1}, f_t, f_{t+1}, f_{t+2}$	f_{t-2}	32.02	32.25	31.18
$f_{t-3}, f_{t-2}, f_t, f_{t+1}, f_{t+2}$	f_{t-1}	**36.23**	37.25	**35.20**
$f_{t-4}, f_{t-2}, f_t, f_{t+2}, f_{t+4}$	f_{t-1}	**36.22**	**37.32**	**35.21**
FastDVDnet superv.	n/a	36.58	37.29	35.82

Table 1: Results for different reference frames and training stacks $\mathcal{S}'_t$. During inference after training is done, we denoise the video using the natural stack $\mathcal{S}_t$. We report the average PSNR of the on-line fine-tuning over 17 sequences from the Derf [40] and Vid3oC-10 [26] datasets, excluding the first 10 frames of each video (to allow for the adaptation time). We considered $3{\times}3$ box noise with $\sigma{=}40$, AWGN $\sigma{=}20$ and scaled Poisson noise with $p = 8$.

Figure 4: Effect of the mask. Errors in the optical flow create mismatches that negatively influence the fine-tuning (*e.g.* transparency of right arm, hallucinated texture). We detect occlusions and warping errors and remove them from the loss via a binary mask.

network output and the desired output y. The optimal estimators $\hat{y} = \mathcal{F}^*(z)$ capture some aspects of the data distribution, which is the point of a data-driven approach. For instance, for the MSE loss we have $\mathcal{F}^*(z) = \mathbb{E}\{y|z\}$ [24].

We would like to configure the input stack to our network so that the fine-tuning can learn such data-driven estimators. The following observation restricts the number of options.

Observation 1. *Let (z, y) distributed according to $p(z, y)$. An estimator $\hat{y}(z) = \mathcal{F}^*(z)$ minimizing the expected value of a loss $\mathbb{E}\{\ell(\mathcal{F}(z), z)\}$ that depends only on z and $\mathcal{F}(z)$ is independent from the data distribution $p(z, y)$.*

As the loss depends only on the input z and $\mathcal{F}(z)$, the minimization of $\mathbb{E}\{\ell(\mathcal{F}(z), z)\}$ can be done for each input z independently, *i.e.* $\mathcal{F}^*(z) = \arg\min_{\hat{y}} \ell(\hat{y}, z)$. Given z, the optimal estimator is the minimizer of the loss for that specific z. As a consequence it is independent of the data distribution and only depends on the chosen loss.

Observation 1 implies that the reference frame cannot be a function of the input stack if we want to have a data-driven estimator. Therefore, for our input stack we will adopt a solution similar to that of blind spot networks [28, 2]: remove the target frame from the input stack. Denoting by $\mathcal{S}'_t$ the fine-tuning stack, we then minimize the following *multi-frame to frame* (MF2F) loss:

$$\ell_1^{\text{MF2F}}\left(\mathcal{F}_\theta(\mathcal{S}'_t), f_{t-1}\right) = \|\kappa_t \circ (W_{t,t-1}\mathcal{F}_\theta(\mathcal{S}'_t) - f_{t-1})\|_1, \tag{3}$$

where the warping operator $W_{t,t-1}$ is defined in (2).

We evaluated the fine-tuning with different configurations of disjoint stack and reference frame: introducing the blind spot at different distances from the central frame or evenly spacing the frames in the stack. Table 1 summarizes the results. We found that: (1) The target frame has to be as close as possible to the denoised frame. Otherwise, the quality of the alignment degrades, negatively impacting the fine-tuning. (2) The results can be slightly improved using the *dilated stack* $[f_{t-4}, f_{t-2}, f_t, f_{t+2}, f_{t+4}]$.

We also observed that, regardless of the fine-tuning stack, the best way to perform the inference is by using the natural stack. Frames in the natural stack are more temporally correlated and this helps improving the denoising. The results reported in Table 1 were obtained using the natural stack for testing. The results obtained using the training stack at testing time (which can be found in the supplementary material) are on average 0.3dB below the ones shown in the table. In our remaining experiments we will always use the dilated stack for fine-tuning and the natural stack for final inference as illustrated in Fig. 3.

2.2. Handling warping errors

Following [18], the warping transformations are estimated using the TV-L1 optical flow method [62, 50], as it gives consistent results across noise types and intensities. Moreover, it is based on minimizing the photometric distance between pixels, which is precisely what we need for our loss.

In Equation (3) the mask κ is zero for regions where a misalignment is likely, and one otherwise. Alignment errors are defined as the union of occlusions computed from the optical flow and regions with a large warping residual

$$r_{t,t-1} = g * \|W_{t,t-1}(g * f_t) - g * f_{t-1}\|_1, \tag{4}$$

where g is Gaussian filter of standard deviation $\sigma = 2$ used to obtain a rough estimate of the clean video. The warping residual is then thresholded with a robust adaptive threshold (see supp. material). This mask differs from the one used in [18], which is based on the divergence of the optical flow.

The mask κ is particularly important for fine-tuning a multi-frame denoising network. Indeed, since the network has access to multiple input frames, it is likely that the mismatched target can be found in the stack. The result in Fig. 4 confirms this. We see that the result of F2F barely changes with or without mask. However, applying MF2F without a mask leads to a degradation of the result.

All the experiments in the following sections were computed using the same parameters for the optical flow and misalignment mask.

		Non-blind			AWGN σ blind		Model blind		
		FastDVDnet supervised			online MF2F	online MF2F	online F2F	online MF2F	offline MF2F
		noise specif.	multi task	VBM3D	scalar sigma	8 sigmas	weights	weights	weights
Derf	Gaussian 20	36.96 / .946	36.49 / .942	36.21 / .933	36.92 / .946	36.90 / .945	34.53 / .909	37.32 / .951	**37.48 / .952**
	Gaussian 40	34.00 / .907	33.70 / .903	32.63 / .871	33.95 / .906	33.91 / .905	32.04 / .847	34.24 / .913	**34.27 / .914**
	Poisson 1	40.45 / .974	39.71 / .970	38.99 / .959	39.50 / .960	40.15 / .972	36.56 / .953	40.39 / **.974**	**40.51 / .974**
	Poisson 8	36.00 / .939	35.58 / .935	34.18 / .897	34.15 / .890	**35.73 / .934**	31.98 / .890	35.57 / .941	35.68 / **.942**
	Box $3 \times 3, 40$	35.42 / .932	34.86 / .924	29.94 / .757	34.13 / .900	34.34 / .902	32.55 / .891	35.51 / **.927**	**35.60 / .927**
	Box $5 \times 5, 65$	34.78 / .932	33.98 / .919	28.37 / .736	32.14 / .873	32.60 / .888	31.78 / .886	34.29 / **.928**	**34.35 / .928**
	Demosaicked 4	34.85 / .926	25.53 / .533	33.16 / .890	33.23 / .877	34.30 / .916	32.61 / .885	34.75 / .926	**34.81 / .927**
Vid3oC-10	Gaussian 20	37.49 / .964	36.84 / .959	36.55 / .951	37.43 / .963	37.40 / .963	31.62 / .868	37.32 / .964	**37.55 / .966**
	Gaussian 40	34.27 / .937	33.70 / .929	32.74 / .901	34.23 / .935	34.21 / .935	29.10 / .799	34.17 / .937	**34.26 / .938**
	Poisson 1	40.63 / .980	39.71 / .975	39.32 / .967	39.30 / .966	**40.29 / .978**	33.54 / .905	40.01 / .978	40.16 / **.979**
	Poisson 8	35.72 / .951	35.09 / .944	32.16 / .844	33.26 / .887	**35.42 / .946**	29.92 / .823	34.99 / **.947**	35.00 / **.947**
	Box $3 \times 3, 40$	37.28 / .964	36.54 / .957	30.19 / .770	34.70 / .936	34.90 / .938	31.27 / .871	36.65 / **.963**	**36.76 / .963**
	Box $5 \times 5, 65$	36.81 / .965	35.84 / .955	28.53 / .746	32.65 / .909	33.11 / .917	30.75 / .869	35.65 / .956	**35.79 / .957**
	Demosaicked 4	34.50 / .941	23.96 / .508	32.31 / .882	32.24 / .876	33.86 / .931	31.59 / .890	33.95 / .933	**33.98 / .934**

Table 2: Average PSNR and SSIM over all the sequences for a given dataset and type of noise. The MF2F fine-tuning is applied to a FastDVDnet network [55], either on the weights (model blind) or the input variance map (σ blind). F2F fine-tuning is applied to the weights of a single frame DnCNN network [64]. The best PSNR in each case is underlined. The best blind method is in **bold**.

2.3. Fine-tuning and inference

Similarly to [18], the proposed fine-tuning can be done online or offline. In the offline setting, the network is first fine-tuned on the entire video, which is considered as a dataset of frames. We form batches by randomly sampling frame stacks from the video and update the network parameters by performing one optimizer step per batch. This is repeated a fixed number of epochs. Afterwards, the fine-tuned parameters are used to denoise the video by applying the network to each frame using the natural input stack.

The online setting defines a time-varying sequence of network parameters, and can thus adapt to temporal changes in the distribution of the noise or the signal. The video is processed sequentially, applying the following two steps on each frame. First the network parameters are updated by performing a fixed number N of optimizer steps of the loss (3). Then the denoised output is produced by applying the updated network on the natural input stack.

The proposed fine-tuning can be applied to the network weights, or any other parameter that has an influence on the denoising performance, such as the variance map Σ_t that FastDVDnet takes as input (see Fig. 2). In FastDVDnet [54] the authors consider only homoscedastic AWGN, and use therefore a constant image $\Sigma_t(x) = \sigma^2$ as variance map. The input variance map controls the denoising strength and the correct value has to be provided at inference time. In Section 3.2 we apply the MF2F fine-tuning to simultaneously estimate the variance map and denoise the video. This is particularly relevant when the noise is AWGN, but its variance map unknown.

In all the experiments in upcoming sections we use the same hyper-parameters when the fine-tuning is done with respect to the weights. In the online setting we use a learning rate of 10^{-5} and $N = 20$ iterations of the Adam opti-

mizer on mini-batches consisting of pairs of frames (*i.e.* the weights are updated each two frames). In the offline setting we use the same learning rate and perform 200 Adam iterations with mini-batches of 20 frames (no improvement was observed with more iterations). When the fine-tuning is applied to the variance map, we adapt the learning rate and keep the same number of iterations and batch sizes.

3. Experiments on synthetic noise

We now present results of the proposed framework for a wide range of synthetic noise types and use cases. The evaluation is performed on videos from two datasets. One is a set of seven Full HD videos of 100 frames each extracted from the Derf's Test Media collection [40]. The second, more challenging, dataset consists of ten videos of 120 frames extracted from the training split of the Vid3oC dataset [26]. We refer to this dataset as Vid3oC-10. We downscaled all videos by a factor of two.

In our experiments we consider four noise types (1) AWGN noise, (2) scaled Poisson noise with scaling parameter p (the mean of the noisy pixel f_i is the clean pixel u_i, and the variance is pu_i), (3) correlated noise (denoted *box noise*) obtained by filtering AWGN with an $s \times s$ box filter and (4) demosaicked Poisson noise, obtained by mosaicking the image, adding scaled Poisson noise and then applying the demosaicking algorithm of [25]. For the first three types we consider two noise levels, indicated in the first columns of Table 2. The demosaicked Poisson noise simulates the correlation introduced by a demosaicking algorithm applied on the noisy data. We evaluate the average PSNR and SSIM for the given sequence using the ground-truth, but excluding the first 10 frames of each sequence. Additional qualitative comparisons of the methods can be found in the supplementary material.

We remark that all results are obtained by fine-tuning a network independently on each video. In all cases the fine-tuning starts from weights pre-trained for AWGN with $\sigma = 25$. The proposed fine-tuning scheme can be applied to any denoising network and any pre-trained weights can be used as starting point. In the supplementary material we studied the impact of the starting point on the performance of the fine-tuned network for different noise types. Unsurprinsingly, the the speed of the adaptation depends on the similarity between the pre-trained and the target noise distributions. We observed that it is easier for the AWGN weights to adapt to other noise types.

3.1. Fine-tuning the network weights

In the following experiments we apply the proposed fine-tuning to the network weights of FastDVDnet, while keeping constant the input variance map. We compare with F2F [18] which is, to the best of our knowledge, the only other blind video denoising method in the literature. As reference, we compare with three non-blind algorithms. (1) A *noise-specific* FastDVDnet network trained in a supervised setting for each kind of noise. For AWGN we used the weights provided by the authors of FastDVDnet [55], which work in conjunction with the input variance map. For other noise types we retrained one FastDVDnet per noise type and intensity, using the same dataset [45] and hyperparameters as in [55]. (2) A *multi-task* FastDVDnet trained to handle multiple noise types: Gaussian ($\sigma = 20$ and $\sigma = 40$), Poisson ($p = 1$ and $p = 8$), and our box noise (3×3, $\sigma = 40$ and 5×5, $\sigma = 65$). (3) The VBM3D [13] path-based method in which the noise parameter is set to yield the best result.

In Table 2, we report quantitative results obtained with the proposed MF2F fine-tuning applied on the network weights both in the online and the offline settings. From the results we can see that the performance of the offline MF2F method is slightly superior to the online one. Compared against the multi-task network the proposed MF2F fine-tuning always attains better results. Furthermore, the multi-task network cannot handle demosaicking noise (which was unseen during training) while our self-supervised networks compete with the network specifically trained with supervision for that noise type.

For most noise types, the results of our self-supervised MF2F fine-tuning are close to those obtained with the noise-specific FastDVDnet network trained with supervision. This is confirmed in Fig. 1, where we compare some results of the offline method with the noise-specific FastDVDnet (supervised), and F2F.

In all experiments we observe a consistent PSNR gain of about 3dB with respect to F2F. This corresponds to a noise reduction of a factor 2, which is expected from a network that exploits the redundancy of 5 frames compared against a single-frame method.
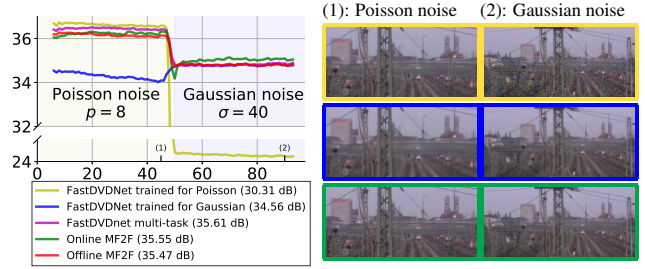

Figure 5: Adaptation to changes in the noise properties. We simulate a sequence with Poisson noise for the first half and Gaussian noise for the second half. The frames (1) and (2) corresponds to Poisson and Gaussian noise respectively. The pretrained methods for the specific noise types perform poorly on the other half (yellow and blue), while the proposed methods (online and offline) are able to cope with the abrupt change.

Time varying noise. The online fine-tuning of the weights permits to quickly adapt to changes in the noise properties. The PSNR plot in Fig. 5 shows the per-frame PSNR computed on a video in which the noise switches from Poisson ($p = 8$) to Gaussian ($\sigma = 40$) at frame 50. While the two noise-specific FastDVDnet networks perform well for their respective noise types, their performance strongly degrades for the other type. On the other hand, the proposed MF2F approaches are able to cope with the abrupt change of noise, and the online version even outperforms the network trained for Gaussian on the Gaussian section. The offline method is able to handle both noise types by learning to denoise them with the same network. The results obtained with MF2F are on par to those of a multi-task FastDVDnet trained with supervision for these two noise distributions. The crops in Fig. 5 show that the noise-specific networks failed for the other noise types (as expected), while the online method is able to restore fine details for both noise types.

3.2. Fine-tuning only the noise map

The proposed fine-tuning can be used to estimate the input variance map of the network, while keeping the network weights fixed. This is done by minimizing the MF2F loss (3) with respect to the variance map $\Sigma_t(x)$. To highlight the flexibility of the framework, we examine three parameterizations of the input variance map. All experiments in these section were performed with the online fine-tuning. Thus, a different input variance map is estimated for each frame, and used as the starting condition for the next frame.

Homoscedastic AWGN. If a sequence has homoscedastic AWGN of unknown variance σ^2, we set a constant variance map $\Sigma_t(x) = \sigma^2$. By fine-tuning only with respect to σ, we obtain a σ-blind denoiser (*scalar sigma* in Table 2). The convergence is very fast and after a few frames the estimated σ stabilizes around the real noise variance. If the noise is not homoscedastic AWGN, a compromise value for σ is found.
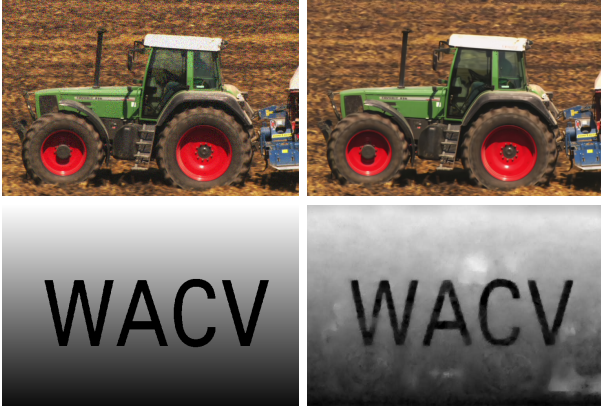
Figure 6: Spatially variant variance map used in a synthetic experiment. From left to right and top to bottom: a noisy frame; the corresponding denoised by the proposed online MF2F; variance map of the added Gaussian noise; and the variance map obtained by using the proposed fine-tuning of the input variance map. The noise level of the letters is $\sigma = 0$ and varies linearly in the background from $\sigma = 0$ to 40;

Heteroscedastic AWGN. We now consider AWGN with a space varying variance map which is constant in time, *i.e.* $\Sigma_t(x) = \Sigma_0(x)$, and fine-tune with respect to the entire input variance map. We add a TV regularization term to the loss (*e.g.* [48]) enforcing smoothness on the variance map. We tested this on a sequence contaminated with AWGN with variance map shown in Fig. 6. Our method is able to recover the message hidden in the noise variance, with remarkable spatial resolution. The convergence took around 40 frames, which is quite slower than for the scalar σ. We are not aware of any other method that can estimate such a noise map from a video. This could be useful in practical cases where the noise distribution is spatially variant, such as optical images with vignetting or MRI imaging [31]. In these cases the variance map is much smoother than the artificial one we used for our experiment. Note that the ultimate goal here is not to accurately recover the noise map, but to maximize the denoising performance. For example, in a flat region, the fine-tuning might lead to an over-estimation of the variance so as to increase the amount of smoothing. This explains the differences in Fig. 6 between the actual noise map and the one found.

Signal dependent AWGN. Poisson noise can be approximated as AWGN with signal dependent variance map given by $\Sigma_t(x) \propto u_t(x)$. To cope with this noise, we use a spatially-variant noise map based on the brightness of the noisy video. We parametrize the variance map with K trainable parameters $\sigma_1, \ldots \sigma_K$. We split the intensity range of the noisy video in K equal intervals, each with a corresponding σ_i. The input variance map at pixel (x, t) is the σ_i corresponding to the pixel brightness. At each frame,

the σ_i are automatically determined with the proposed online MF2F fine-tuning. We set $K = 8$, since it results a good trade-off between efficiency and denoising performance. Quantitative results are reported in Table 2. For Poisson and AWGN noise, the obtained results are close to the ones obtained by the noise-specific FastDVDnet trained with supervision. Visual results for Poisson noise can be found in the supplementary material.

4. Results on real noisy videos

In this section we show results on real noisy videos. In these examples the network adapts not only to different types of noise, but also to signals on different domains.

Fig. 7 shows results obtained on the outdoor CRVD dataset [61]. This dataset consists in raw sequences of 50 frames with real noise from a surveillance camera with the sensor IMX385, at five ISO levels. Since our fine-tuning does not handle mosaicked videos, we first applied a simple tone curve ($\gamma = 2.5$) and a demosaicking algorithm [25]. We then applied our offline blind denoising on the demosaicked noisy frames. Although the conventional approach is to perform first denoising and then demosaicking (or even better, perform both jointly), recent work in [22] suggested that good results can also be obtained by applying the denoising after the demosaicking. We compare with the multi-frame denoiser of RViDeNet [61] which was fine-tuned on the indoor CRVD dataset, consisting on short raw stop motion sequences with ground truth. This network takes raw noisy videos as input. In order to obtain comparable results, the denoised raw frames are then tone-mapped and demosaicked [25] as described above. From Fig. 7 we can see that the results of MF2F are sharper and contain more details than those of RViDeNet, and has less residual noise than F2F (more visual results for the five ISO levels can be found in the supplementary material).

To obtain a quantitative evaluation on realistic noisy videos we follow [61] and use the *unprocessing* network [4] to simulate raw videos from clean RGB ones. These simulated videos are also used in [61] to pre-train (in a supervised setting) RViDeNet. We consider the Poisson-Gaussian noise (to model the shot and read noises), using the parameters estimated in [61] for the ISO levels 1600, 3200, 6400, 12800 and 25600. To avoid any influence from the demosaicking step we evaluate the performance on the raw denoised videos. For MF2F the fine-tuning is performed independently on each video. We apply the same raw process as for the outdoor CRVD dataset: tone curve and demosaicking [25] before denoising. Since the results are in sRGB, we re-mosaick them and invert the tone curve for evaluating the PSNR in the raw domain. Table 3 presents the average PSNRs on all the sequences for each ISO level. We can see that even though RViDeNet was pre-trained on the same noise type, MF2F performs much better.
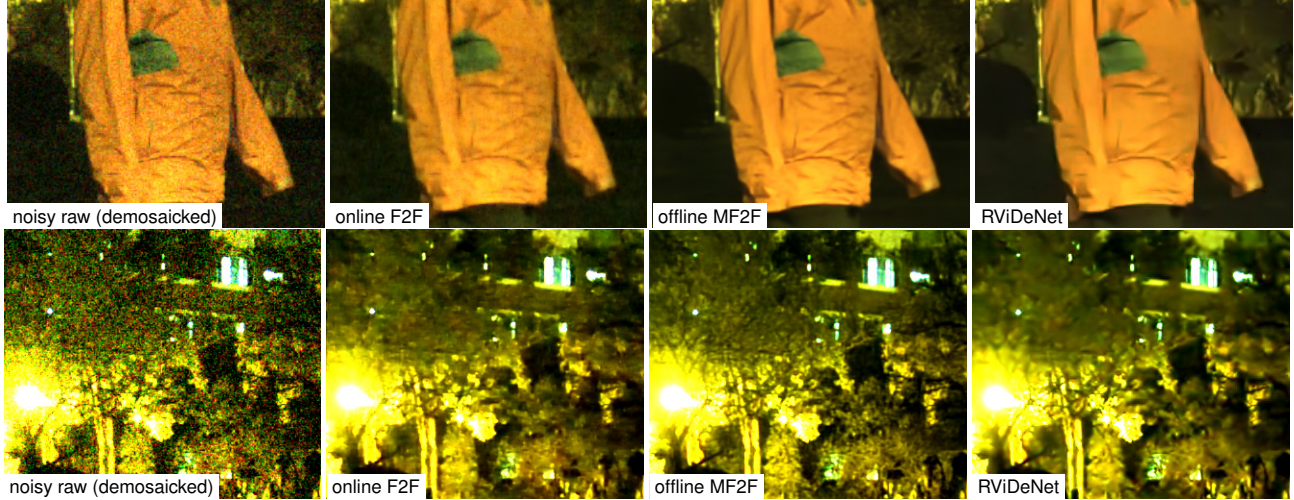
Figure 7: Details from frame of a denoised raw video (ISO 12800) processed by F2F, offline MF2F, and RViDeNet. All images are demosaicked and gamma corrected.

	1600	3200	6400	12800	25600
RViDeNet	45.79	44.26	42.64	40.83	38.93
Online MF2F	**46.51**	**45.13**	**43.64**	**41.91**	**39.68**

Table 3: Synthetic raw denoising: average PSNR per ISO levels over the dataset Derf. The best PSNR in each case is in **bold**.
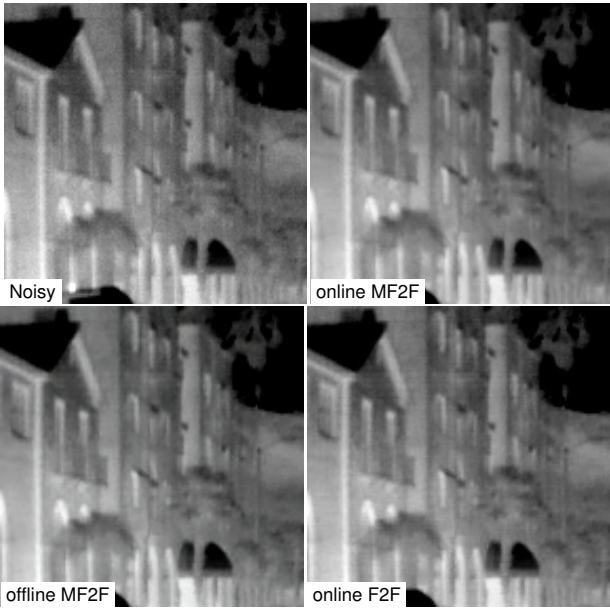
Figure 8: Example frame of a restoration of infra-red real video.

Lastly, we also tested MF2F on thermal infra-red video from the FLIR ADAS [21] dataset, which consists of videos taken with a Tau2-640 sensor. Fig. 8 shows a frame from the result obtained with the offline MF2F method on a sequence of 4224 frames.

5. Conclusions

In this work we address the problem of blind video denoising. To that aim, we extend the self-supervised fine-tuning approach introduced in [18] to multi-frame denoising networks. This is achieved by fine-tuning the network using a dilated input frame stack and switching back to the natural input stack at inference time.

The proposed approach demonstrates that by exploiting the temporal consistency in videos it is possible to fine-tune a video denoising network using only a few frames of a single noisy sequence and attain the performance of a network trained with supervision on a large dataset. This also allows to handle time-varying noise, which could be useful for vision systems exposed to varying conditions (for instance a surveillance camera at day and night).

There are some interesting future research perspectives. For instance, *meta-learning* could be used to improve the adaptation speed of the pre-trained network [56]. Also, the optical flow and the warping mask used in the MF2F loss could be computed by a network and trained at the same time, similar to [60, 16].

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Journal of Machine Learning Research*, pages 524–533, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[3] Jérôme Boulanger, Charles Kervrann, Patrick Bouthemy, Peter Elbau, Jean-Baptiste Sibarita, and Jean Salamero. Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. *IEEE Transactions on Medical Imaging*, 29(2):442–454, 2009.

[4] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11036–11045, June 2019.

[5] Benoit Brummer and Christophe De Vleeschouwer. Natural image noise dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2019.

[6] Yi Chang, Luxin Yan, Meiya Chen, Houzhang Fang, and Sheng Zhong. Two-stage convolutional neural network for medical noise removal via image decomposition. *IEEE Transactions on Instrumentation and Measurement*, pages 1–1, 2019.

[7] Chen Chen, Qifeng Chen, Minh N. Do, and Vladlen Koltun. Seeing motion in the dark. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[8] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3155–3164, June 2018.

[10] Xinyuan Chen, Li Song, and Xiaokang Yang. Deep rnns for video denoising. In *Applications of Digital Image Processing*, 2016.

[11] Michele Claus and Jan van Gemert. Videnn: Deep blind video denoising. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, June 2019.

[12] Pierrick Coupé, Pierre Hellier, Charles Kervrann, and Christian Barillot. Nonlocal means-based speckle filtering for ultrasound images. *IEEE Transactions on Image Processing*, 18(10):2221–2229, 2009.

[13] Kostadin Dabov, Alessandro Foi, and Karen Egiazarian. Video denoising by sparse 3D transform-domain collaborative filtering. In *EUSIPCO*, 2007.

[14] Axel Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. A Non-Local CNN for Video Denoising. In *The IEEE International Conference on Image Processing (ICIP)*, volume 2019-Septe, pages 2409–2413. IEEE, sep 2019.

[15] Axel Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. Video Denoising by Combining Patch Search and CNNs. *Journal of Mathematical Imaging and Vision*, pages 1–16, oct 2020.

[16] Michel Deudon, Alfredo Kalaitzis, Md Rifat Arefin, Israel Goytom, Zhichao Lin, Kris Sankaran, Vincent Michalski, Samira E Kahou, Julien Cornebise, and Yoshua Bengio. Highres-net: Multi-frame super-resolution by recursive fusion. Technical report, 2020.

[17] Thibaud Ehret, Axel Davy, Pablo Arias, and Gabriele Facciolo. Joint demosaicing and denoising by overfitting of bursts of raw images. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[18] Thibaud Ehret, Axel Davy, Jean-Michel Morel, Gabriele Facciolo, and Pablo Arias. Model-blind video denoising via frame-to-frame training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Mario Gonzalez, Javier Preciozzi, Pablo Muse, and Andres Almansa. Joint denoising and decompression using cnn regularization. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018.

[20] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. *arXiv preprint arXiv:1807.04686*, 2018.

[21] FLIR Systems Inc. Flir thermal dataset for algorithm training. *Online, https://www.flir.in/oem/adas/adas-dataset-form/*, 2018.

[22] Qiyu Jin, Gabriele Facciolo, and Jean-Michel Morel. A review of an old dilemma: Demosaicking first, or denoising first? In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 514–515, June 2020.

[23] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical physics*, 44(10):e360–e375, 2017.

[24] Steven M Kay. Fundamentals of statistical processing, volume i: Estimation theory: Estimation theory v. 1, 1993.

[25] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Minimized-Laplacian residual interpolation for color image demosaicking. In Nitin Sampat, Radka Tezaur, Sebastiano Battiato, and Boyd A. Fowler, editors, *Digital Photography X*, volume 9023, pages 197 – 204. International Society for Optics and Photonics, SPIE, 2014.

[26] Sohyeong Kim, Guanju Li, Dario Fuoli, Martin Danelljan, Zhiwu Huang, Shuhang Gu, and Radu Timofte. The vid3oc and intvid datasets for video super resolution and quality mapping. In *The International Conference on Computer Vision Workshop (ICCVW)*, pages 3609–3616. IEEE, 2019.

[27] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer Learning From Synthetic to Real-Noise Denoising With Adaptive Instance Normalization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3479–3489. IEEE, jun 2020.

[28] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void - Learning Denoising From Single Noisy Images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2124–2132. IEEE, jun 2019.

[29] Alexander Krull, Tomas Vicar, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. Technical report, 2019.

[30] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*. 2019.

[31] Bennett A Landman, Pierre-Louis Bazin, Seth A Smith, and Jerry L Prince. Robust estimation of spatially variable noise fields. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 62(2):500–509, 2009.

[32] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The noise clinic: a blind image denoising algorithm. *Image Processing On Line*, 5:1–54, 2015.

[33] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Journal of Machine Learning Research*, pages 2965–2974. PMLR, 10–15 Jul 2018.

[34] Ce Liu and William T. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *European Conference on Computer Vision (ECCV)*, pages 706–719, 2010.

[35] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[36] Matteo Maggioni, Enrique Sánchez-Monge, and Alessandro Foi. Joint removal of random and fixed-pattern noise through spatiotemporal video filtering. *IEEE Transactions on Image Processing*, 23(10):4282–4296, 2014.

[37] Talmaj Marinč, Vignesh Srinivasan, Serhan Gül, Cornelius Hellge, and Wojciech Samek. Multi-kernel prediction networks for denoising of burst images. In *The IEEE International Conference on Image Processing (ICIP)*, pages 2404–2408, September 2019.

[38] Christopher A. Metzler, Ali Mousavi, Reinhard Heckel, and Richard G. Baraniuk. Unsupervised learning with stein's unbiased risk estimator, 2018.

[39] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] Chris Montgomery et al. Xiph. org video test media (derf's collection), the xiph open source community, 1994. *Online, `https://media.xiph.org/video/derf`*.

[41] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2Noise: Learning to Denoise from Unpaired Noisy Data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12064–12072, June 2020.

[42] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[43] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1586–1595, July 2017.

[44] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[45] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[46] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2Self With Dropout: Learning Self-Supervised Denoising From Single Image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1890–1898, June 2020.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Miccai*, pages 234–241, 2015.

[48] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[49] Joseph Salmon, Zachary Harmany, Charles-Alban Deledalle, and Rebecca Willett. Poisson noise reduction with non-local pca. *Journal of mathematical imaging and vision*, 48(2):279–294, 2014.

[50] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. *Image Processing On Line*, 2013:137–150, 2013.

[51] Venkataraman Santhanam, Vlad I. Morariu, and Larry S. Davis. Generalized deep image to image regression. *CoRR*, abs/1612.03268, 2016.

[52] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3257–3267. Curran Associates, Inc., 2018.

[53] Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, 9(6):1135–1151, 11 1981.

[54] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *The IEEE International Conference on Image Processing (ICIP)*, September 2019.

[55] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1354–1363, June 2020.

[56] Alessio Tonioni, Oscar Rahnama, Tom Joy, Luigi Di Stefano, Ajanthan Thalaiyasingam, and Philip Torr. Learning to adapt for stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[57] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, Dec. 2010.

[58] Puyang Wang, He Zhang, and Vishal M Patel. Sar image despeckling using a convolutional neural network. *IEEE Signal Processing Letters*, 24(12):1763–1767, 2017.

[59] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, Aug 2019.

[60] Songhyun Yu, Bumjun Park, Junwoo Park, and Jechang Jeong. Joint learning of blind video denoising and optical flow estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 500–501, June 2020.

[61] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2301–2310, June 2020.

[62] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*. Springer, 2007.

[63] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2696–2705, June 2020.

[64] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 7 2017.

[65] Weiying Zhao, Charles-Alban Deledalle, Loïc Denis, Henri Maître, Jean-Marie Nicolas, and Florence Tupin. Ratio-based multitemporal sar images denoising: Rabasar. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):3552–3565, 2019.