Unsupervised Deep Video Denoising with Untrained Network

Huan Zheng,[1] **Tongyao Pang,** [1] **Hui Ji** [1]

[1] Department of Mathematics at National University of Singapore, Singapore.
huan_zheng@u.nus.edu, matpt@nus.edu.sg, matjh@nus.edu.sg

Abstract

Deep learning has become a prominent tool for video denoising. However, most existing deep video denoising methods require supervised training using noise-free videos. Collecting noise-free videos can be costly and challenging in many applications. Therefore, this paper aims to develop an unsupervised deep learning method for video denoising that only uses a single test noisy video for training. To achieve this, an unsupervised loss function is presented that provides an unbiased estimator of its supervised counterpart defined on noise-free video. Additionally, a temporal attention mechanism is proposed to exploit redundancy among frames. The experiments on video denoising demonstrate that the proposed unsupervised method outperforms existing unsupervised methods and remains competitive against recent supervised deep learning methods.

1 Introduction

Despite advances in optical technology, noise remains a common degradation source for images, especially when captured by compact devices or in high-sensitivity settings such as low lighting or high frame rates. Denoising is an important preprocessing step in many computer vision applications, as its performance can impact the accuracy of subsequent processes. Videos are a rich source of data with numerous applications, but video denoising differs from image denoising in several aspects. Although videos can be thought of as sequences of images, video frames often have lower signal-to-noise ratio (SNR) than individual images, due to the faster shutter speed required for video capture. Additionally, there exist high redundancies among adjacent frames in videos, providing multiple noisy instances of the same image pixel. As a result, effective and efficient exploitation of temporal redundancy is a focus of video denoising to achieve better performance than single-image denoising

Motivation

In the past few years, deep learning has emerged as a powerful tool for video denoising. The majority of existing deep learning methods for video denoising are based on supervised learning, as seen in works such as (Davy et al. 2019; Tassano, Delon, and Veit 2020, 2019; Maggioni et al. 2021; Li et al. 2022). While these methods differ in terms of network architectures or training schemes, they all train their networks using numerous paired training samples consisting of both noisy and noise-free videos. However, collecting a large-scale dataset with noise-free videos for dynamic scenes, as well as some applications such as dynamic medical imaging and microscopy, can be both costly and challenging. Additionally, the introduction of bias from training samples can be a concern in practice.

Recently, there has been increasing interest in studying unsupervised methods for video denoising, which do not require noise-free videos for training. F2F (Ehret et al. 2019) extended Noise2Noise (Lehtinen et al. 2018), a semi-supervised image denoiser, to frame-to-frame video denoising. Motivated by the redundancy in adjacent frames, MF2F (Dewil et al. 2021) imposed a loss on multiple aligned frames for improved performance. RFR (Lee et al. 2021) utilized a pre-trained denoiser to synthesize pseudo-clean videos for simulating supervised learning. UDVD (Sheth et al. 2021) extended Blind Spot (Batson and Royer 2019; Krull, Buchholz, and Jug 2019; Laine et al. 2019), an unsupervised image denoising network architecture, to video denoising using a bias-free network. However, unsupervised methods for video denoising are still in their infancy, and there is still a noticeable performance gap between existing unsupervised methods and their supervised counterparts.

This paper aims to develop an unsupervised deep learning method for video denoising using an untrained deep network. The proposed method achieves state-of-the-art performance among both unsupervised and supervised video denoising methods and does not require access to any external training samples with noise-free videos.

Main contribution

To develop an unsupervised deep learning method for video denoising, there are two questions to address: (a) how to design a loss function such that one can train the network without calling any noise-free videos; (b) how to effectively exploit temporal redundancy among adjacent frames in the presence of possible misalignment errors.

Our answer to Question (a) is a self-supervised loss that provides a good estimation of the loss function defined over noise-free videos with mathematical justification. This loss is inspired by the R2R method (Pang et al. 2021), a self-

supervised loss defined over a pair of images constructed from one single noisy image by a specific scheme. The scheme proposed in (Pang et al. 2021) is only applicable to Gaussian noise. This paper further extended the scheme of R2R to the case where measurement noise can be more varied and not necessarily Gaussian. This paper showed that the resulting loss function remains a good estimation of the loss defined over the pair of noisy/truth images. Based on the proposed extended R2R (ER2R) loss for video denoising, we first train a spatial denoising module over noisy frames for facilitating frame alignment. Afterwards, a video ER2R (VER2R) loss is introduced to guide the training of the denoising network in the absence of training samples with noise-free videos.

Our answer to Question (b) is a temporal attention mechanism for effectively and efficiently exploiting the redundancy of image pixels in their temporal neighbourhoods. The attention mechanism has been utilized in some existing supervised video denoising networks. ST-PAN (Xu et al. 2020) and KPN (Mildenhall et al. 2018) learned a 3D spatial-temporal attention weight/kernel to aggregate the neighbouring pixels in spatial-temporal domain. As such, 3D attention blocks lead to very high computational cost. RViDeNet (Yue et al. 2020a) and BPN (Xia et al. 2020) improved computational efficiency by either adopting a separate and parallel attention module, or compacting the kernel space by a linear space over some learned bases.

To account for alignment errors, we incorporate a residual neural network for motion correction and introduce a lightweight temporal attention module for predicting the weights related to the correlation of the target pixel and its neighbouring pixels. These weights are used for fusing the temporal neighbours of the target pixel for denoising. Such a lightweight attention mechanism makes the proposed method computationally efficient yet provides competitive performance. See below for the summary of the contributions of this paper.

- **A self-supervised loss function for general random noise**. With mathematical justification, a self-supervised loss function without accessing noise-free data is proposed for approximating the supervised loss defined over noise-free data.

- **A light-weight temporal attention module for exploiting temporal redundancy of video frames.** Combining with the proposed self-supervised loss, a simple temporal NN with a light-weight temporal attention mechanism is developed for effectively exploit temporal redundancy in video while remaining computationally efficient.

- **An unsupervised video denoising network with competitive performance.** The proposed method not only outperformed existing unsupervised denoising methods, but also remained very competitive against recent supervised deep learning methods.

2 Related work

Unsupervised deep learning methods for image denoising.

Classical image denoising methods impose a pre-defined image prior on truth images for regularizing the denoising process. For example, the TV method (Rudin, Osher, and Fatemi 1992) imposed a sparsity prior on image gradients for image denoising. The BM3D method (Dabov et al. 2007) imposed a recurrence prior on image patches. Such a regularization approach can also be integrated with representation learning (see *e.g.* (Aharon, Elad, and Bruckstein 2006; Cai et al. 2014)) or ensemble learning (Yang et al. 2020).

In recent years, deep learning has become a prominent tool for image denoising. While supervised deep learning methods have shown impressive performance, their prerequisite of training samples with noise-free images limits their practical usage. Recently, some unsupervised/self-supervised deep denoisers have been proposed to relax this prerequisite.One early work is deep image prior (DIP) (Ulyanov, Vedaldi, and Lempitsky 2018) which shows the implicit regularization induced by a CNN. Another pioneering work is Noise2Noise (N2N) (Lehtinen et al. 2018) which allows one to replace noisy/clean image pairs by independent noisy/noisy pairs for training. Furthermore, the blind-spot relating methods (Krull, Buchholz, and Jug 2019; Batson and Royer 2019; Laine et al. 2019) can effectively train the denoising network on unpaired noisy images, where the main idea is to predict the centering pixel using its neighbours which can alleviate the overfitting. Based on the assumption of Gaussian noise. Self2Self (S2S) (Quan et al. 2020) proposed to use a dropout network for training and inference on a single noisy image. Recorrupted2Recorrupted (R2R) (Pang et al. 2021) proposed a construction scheme for synthesizing a pair of noisy/noisy images which simulates well the pair in N2N such that one can train a denoising network on a set of only noisy images.

Supervised deep learning methods for video denoising.

The study on video denoising is currently dominated by supervised deep learning. As there is a strong correlation between different frames in videos, an effective video denoising method should utilize temporal correlation to achieve better performance than independently denoising each image frame. Based on a pre-processing step of patch matching which stacks together similar image patches in a spatial-temporal neighborhood, VNLNet(Davy et al. 2019) fed matched image patches into an image denoising network, DnCNN (Zhang et al. 2017). PaCNet (Vaksman, Elad, and Milanfar 2021) is also patch-based and it improves efficiency by introducing the concept of patch-craft frames. These frames are synthesized using nearest neighbors in a spatial-temporal window and then augmented to video frames for denoising. However, these patch-based networks have a high computational cost as they extensively call patch matching.

The spatial-temporal correlation of video frames can also be exploited by specific neural network designs. DVD-

Net (Tassano, Delon, and Veit 2019) first runs a spatial denoising block for motion compensation by estimating optical flow and then feeds the aligned frames to a temporal denoising block for fusion. FastDVDNet (Tassano, Delon, and Veit 2020) modifies DVDNet by using multi-scale U-Nets as denoising blocks without explicitly estimating optical flow, leading to better performance and less computational cost. ST-PAN (Xu et al. 2020) develops an attention-based network to aggregate spatial-temporal pixels using an offset network to sample pixels and an attention NN to predict weights for those sampled pixels. RViDeNet (Yue et al. 2020a) aims to denoise raw videos and contributes a dynamic raw video dataset with noisy-clean pairs. They separately denoise RGBG channels and finally fuse these four channels to form a denoised video. EMVD (Maggioni et al. 2021) proposes a recurrent multi-stage neural network for video denoising with much lower complexity. BiRNN (Chan et al. 2021) proposes to use bidirectional recurrent modules for information propagation. FloRNN (Li et al. 2022) improves computational efficiency by using only lookahead recurrent modules, not backward recurrent modules.

Unsupervised deep learning methods for video denoising.

The non-learning methods for video denoising include VBM4D (Maggioni et al. 2011), which extends BM3D from image denoising to video denoising by searching for similar patches in a spatial-temporal volume. VNLB (Arias and Morel 2018) is another patch-based method that models each group of similar patches as a Gaussian distribution and employs empirical Bayesian estimation. Recently, some unsupervised deep learning methods for image denoising have been extended to video denoising. Frame2Frame (F2F) (Ehret et al. 2019) registers consecutive frames using the optical flow estimated by TV-L1 (Pérez, Meinhardt-Llopis, and Facciolo 2013) and treats aligned frames as independent noisy realization of the same clean image. Then they are used to fine-tune a pre-trained denoising network using N2N. The work *et al.* (Yu et al. 2020) and Multi-Frame2Frame (MF2F) (Dewil et al. 2021) are also based on N2N. The former designs a flow estimation module, which is jointly trained with the denoising module. The later extends F2F from single frame to multi-frame. Unsupervised Deep Video Denoiser (UDVD) (Sheth et al. 2021) extended the blind-spot technique for image denoising to video denoising, which takes several consecutive noisy frames as input and produces a denoised centering frame as output. In addition, the bias-free network used in UDVD implicitly introduces motion compensation. The restore-from-restored (RFR) method (Lee et al. 2021) employs a pre-trained video denoising network to synthesize the pairs of pseudo clean and noisy video frames, which are then used to fine-tune the denoising network.

3 Main Body

This section is devoted to a detailed discussion of the proposed unsupervised video denoising method. Figure 1 shows the pipeline of the proposed method. The training takes a two-stage approach: (i) Stage one: pre-training the spatial denoiser $\mathcal{S}_{\boldsymbol{\theta}^s}$ using the ER2R loss $\mathcal{L}^s(\boldsymbol{\theta}^s)$, which gives a pre-trained model $\mathcal{S}_{\bar{\boldsymbol{\theta}}^s}$; (ii) Stage two: training the whole NN using the VER2R loss $\mathcal{L}^{all}(\boldsymbol{\theta})$. The proposed network architecture consists of three modules: one spatial denoising module for pre-processing, one motion-compensation module for handling alignment errors, and a temporal fusion module with attention mechanism for refining the estimation of video frames.

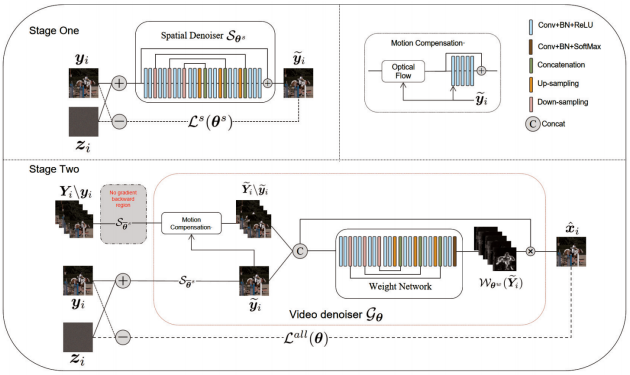

Figure 1: The pipeline of the proposed method. The input volume $\boldsymbol{Y}_i$ consists of five consecutive frames centering at $\boldsymbol{y}_i$, while $\boldsymbol{Y}_i \backslash \boldsymbol{y}_i$ and $\widetilde{\boldsymbol{Y}}_i \backslash \widetilde{\boldsymbol{y}}_i$ contains only four neighbouring frames without the centering one; $\boldsymbol{z}_i$ denotes the simulated random noise which follows the same distribution as the measurement noise in $\boldsymbol{y}_i$.

ER2R self-supervised loss function

In the absence of noise-free data, it is necessary to design a loss function that can accurately measure the prediction error of the network. In the following section, we present a self-supervised loss function defined only on noisy image frames. This loss function is inspired by R2R (Pang et al. 2021) for removing Gaussian white noise from noisy images. The ER2R loss goes one step further to deal with more general random noise. We first define the ER2R loss for a single image and then extend it to multiple frames.

Let $\mathcal{F}_{\boldsymbol{\theta}}$ denote an NN parametrized by $\boldsymbol{\theta}$. Consider the model $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{y}$ is the noisy image, $\boldsymbol{x}$ the truth and $\boldsymbol{n}$ measurement noise. The ER2R scheme re-corrupts $\boldsymbol{y}$ to generate image pairs $\{\boldsymbol{y} + \boldsymbol{z}, \boldsymbol{y} - \boldsymbol{z}\}$, where $\boldsymbol{z}$ is independently simulated from the same noise distribution as $\boldsymbol{n}$. Then, we define the ER2R loss by

$$\ell_{ER2R}(\boldsymbol{\theta}; \boldsymbol{y}, \mathcal{F}_{\boldsymbol{\theta}}) := \mathbb{E}_{\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y} + \boldsymbol{z}) - (\boldsymbol{y} - \boldsymbol{z})\|_2^2. \quad (1)$$

The ER2R loss (1) indeed is a good estimator of the supervised loss defined on image pairs $\{\boldsymbol{y} + \boldsymbol{z}, \boldsymbol{x}\}$.

Theorem 3.1. *Consider $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$. Assume that conditioned on $\boldsymbol{x}$, $\boldsymbol{n}$ and $\boldsymbol{z}$ are independent and identically distributed (i.i.d.) noise. Then it holds that*

$$\mathbb{E}_{\boldsymbol{y}} \ell_{ER2R}(\boldsymbol{\theta}; \boldsymbol{y}, \mathcal{F}_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{n},\boldsymbol{z}} \|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y} + \boldsymbol{z}) - \boldsymbol{x}\|_2^2 + const. \quad (2)$$

Proof. We can rewrite $\mathbb{E}_{\boldsymbol{y}}\ell_{ER2R}(\boldsymbol{\theta};\boldsymbol{y},\mathcal{F}_{\boldsymbol{\theta}})$ as

$$\begin{aligned}\mathbb{E}_{\boldsymbol{y}}\ell_{ER2R}(\boldsymbol{\theta};\boldsymbol{y},\mathcal{F}_{\boldsymbol{\theta}}) &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{n},\boldsymbol{z}}\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}+\boldsymbol{z})-(\boldsymbol{y}-\boldsymbol{z})\|_2^2 \\ &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{n},\boldsymbol{z}}\|\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}+\boldsymbol{z})-\boldsymbol{x}\|_2^2 + \mathbb{E}_{\boldsymbol{n},\boldsymbol{z}}\|\boldsymbol{n}-\boldsymbol{z}\|^2 \\ &\quad + 2\mathbb{E}_{\boldsymbol{x},\boldsymbol{n},\boldsymbol{z}}\left((\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}+\boldsymbol{z})-\boldsymbol{x})(\boldsymbol{n}-\boldsymbol{z})\right)\end{aligned} \tag{3}$$

The second term $\mathbb{E}_{\boldsymbol{n},\boldsymbol{z}}\|\boldsymbol{n}-\boldsymbol{z}\|^2$ in (3) is a constant irrelevant to $\boldsymbol{\theta}$. Then, the remaining is to prove the third term in (3) vanishes. Note that $\boldsymbol{n}$ and $\boldsymbol{z}$ are i.i.d. conditioned on $\boldsymbol{x}$, we have $\mathbb{E}_{\boldsymbol{x},\boldsymbol{n},\boldsymbol{z}}(\cdot)=\mathbb{E}_{\boldsymbol{x}}(\mathbb{E}_{(\boldsymbol{n},\boldsymbol{z})|\boldsymbol{x}}(\cdot))=\mathbb{E}_{\boldsymbol{x}}(\mathbb{E}_{\boldsymbol{n}|\boldsymbol{x}}\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}}(\cdot))$ and the conditional distribution satisfies $p_{\boldsymbol{n}|\boldsymbol{x}}(\cdot)=p_{\boldsymbol{z}|\boldsymbol{x}}(\cdot)$. Denote $p_0(\cdot):=p_{\boldsymbol{n}|\boldsymbol{x}}(\cdot)=p_{\boldsymbol{z}|\boldsymbol{x}}(\cdot)$. Switching the notation $\boldsymbol{n}$ and $\boldsymbol{z}$, we can obtain

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}}\mathbb{E}_{\boldsymbol{n}|\boldsymbol{x}}\big((\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{n}+\boldsymbol{z})-\boldsymbol{x})\boldsymbol{n}\big) \\ =&\int\big((\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{n}+\boldsymbol{z})-\boldsymbol{x})\boldsymbol{n}\big)p_0(\boldsymbol{n})p_0(\boldsymbol{z})d\boldsymbol{n}d\boldsymbol{z} \\ =&\int\big((\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{z}+\boldsymbol{n})-\boldsymbol{x})\boldsymbol{z}\big)p_0(\boldsymbol{z})p_0(\boldsymbol{n})d\boldsymbol{n}d\boldsymbol{z} \\ =&\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}}\mathbb{E}_{\boldsymbol{n}|\boldsymbol{x}}\big((\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{n}+\boldsymbol{z})-\boldsymbol{x})\boldsymbol{z}\big),\end{aligned}$$

Thus, we have

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x},\boldsymbol{n},\boldsymbol{z}}\big(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{y}+\boldsymbol{z})-\boldsymbol{x})(\boldsymbol{n}-\boldsymbol{z})\big) \\ &=\mathbb{E}_{\boldsymbol{x}}\Big(\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}}\mathbb{E}_{\boldsymbol{n}|\boldsymbol{x}}\big((\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{n}+\boldsymbol{z})-\boldsymbol{x})(\boldsymbol{n}-\boldsymbol{z})\big)\Big)=0.\end{aligned}$$

The proof completes. □

Remark 3.1.1. *Both ER2R and R2R (Pang et al. 2021) take a recorrputed-to-recorrupted scheme for defining the loss. The mathematical justification of R2R calls the statistical property of Gaussian noise. In contrast, the mathematical justification of proposed ER2R loss is based on the symmetry of the original noise and the injected noise for connecting it to the supervised counterpart. Thus, different from that of R2R, the justification of ER2R is applicable to more general noise.*

Extension of ER2R from image to video

In our video denoising pipeline, we use the ER2R loss twice: once for pre-training the spatial module and another for training the entire video denoising NN. The spatial module is denoted by $\mathcal{S}_{\boldsymbol{\theta}^s}$, while the entire network is denoted by $\mathcal{G}_{\boldsymbol{\theta}}$. The spatial module processes each frame independently using the same weights. On the other hand, the entire video denoising NN takes in multiple adjacent frames to leverage the temporal redundancy and outputs the denoised version of each center frame. The details of the NN architecture are discussed in the next section.

Recall that our aim is to recover the clean video frames $\boldsymbol{x}_i$ $(i=1,2,\ldots,T)$ from their noisy version

$$\boldsymbol{y}_i=\boldsymbol{x}_i+\boldsymbol{n}_i, \tag{4}$$

where the measurement noise $\boldsymbol{n}_i$ is assumed to be independent across frames.

Stage one: pretraining of $\mathcal{S}_{\boldsymbol{\theta}^s}$. In our approach, we align video frames using optical flow to exploit temporal redundancy across frames efficiently. However, estimating optical flow can be challenging in the presence of high levels of image noise. Therefore, we first pretrain a spatial denoising network using the unsupervised ER2R loss over noisy frames in the video. This process produces denoised frames that can be more accurately aligned using optical flow methods. The loss function for the spatial denoising module is given by:

$$\mathcal{L}^s(\boldsymbol{\theta}^s)=\frac{1}{T}\sum_{i=1}^{T}\ell_{ER2R}(\boldsymbol{\theta}^s;\boldsymbol{y}_i,\mathcal{S}_{\boldsymbol{\theta}^s}), \tag{5}$$

where $\boldsymbol{y}_i(i=1,2,\ldots,T)$ denote the frames in the noisy video. Note that $\mathcal{L}^s(\boldsymbol{\theta}^s)$ can be viewed as an empirical approximation to the expectation $\mathbb{E}_{\boldsymbol{y}}\ell_{ER2R}(\boldsymbol{\theta}^s;\boldsymbol{y},\mathcal{S}_{\boldsymbol{\theta}^s})$. According to Theorem 3.1 and the central limit theorem, $\mathcal{L}^s(\boldsymbol{\theta}^s)$ is equivalent to the supervised loss up to a constant (which does not affect the NN training) as $T\to\infty$, if $\boldsymbol{y}_i(i=1,2,\ldots)$ are i.i.d. samples from $p(\boldsymbol{y})$. This implies that the self-supervised training of $\mathcal{S}_{\boldsymbol{\theta}^s}$ using the ER2R loss can closely approximate supervised training when the test video has a sufficiently large number of frames.

Stage two: training of the video denoising NN $\mathcal{G}_{\boldsymbol{\theta}}$. Consider each centering reference frame $\boldsymbol{y}_i$. We use its $2t_0+1$ adjacent frames, including itself, $\boldsymbol{y}_{-t_0+i},\ldots,\boldsymbol{y}_i,\ldots,\boldsymbol{y}_{t_0+i}$, to denoise it. For the centering frame $\boldsymbol{y}_i$ itself, we can employ the ER2R scheme directly to generate the training frame pairs $(\boldsymbol{y}_i+\boldsymbol{z}_i,\boldsymbol{y}_i-\boldsymbol{z}_i)$, where $\boldsymbol{z}_i$ is i.i.d. noise to $\boldsymbol{n}_i$. For the other neighboring frames, since the noise in them is independent of both $\boldsymbol{n}_i$ and $\boldsymbol{z}_i$, it does not affect the symmetry between $\boldsymbol{n}_i$ and $\boldsymbol{z}_i$, which is the key point for establishing a link with the supervised loss. Applying ER2R to video denoising, we design an unsupervised loss function for each centering frame $\boldsymbol{y}_i$, where $\boldsymbol{y}_{-t_0+i},\ldots,\boldsymbol{y}_i+\boldsymbol{z}_i,\ldots,\boldsymbol{y}_{t_0+i}$ are used as input (where only the centering frame $\boldsymbol{y}_i$ is recorrupted) and $\boldsymbol{y}_i-\boldsymbol{z}_i$ is used as the target.

We use the bold capital letters to denote volumes, with a subscript indicating the centering frame index and a superscript as the total number of frames in the volume. For example, $\boldsymbol{Y}_i^{2t_0+1}$ denotes the the stack of frames $\{\boldsymbol{y}_{-t_0+i},\ldots,\boldsymbol{y}_i,\ldots,\boldsymbol{y}_{t_0+i}\}$. The subscript and superscript are omitted for simplicity unless necessary. A convenient abuse of notation $\boldsymbol{Y}+\boldsymbol{z}$ represents only adding $\boldsymbol{z}$ to the centering frame $\boldsymbol{y}$ in the volume $\boldsymbol{Y}$. For each $\boldsymbol{Y}$ centred at $\boldsymbol{y}$, our video ER2R (VER2R) loss is defined by

$$\ell_{VER2R}(\boldsymbol{\theta};\boldsymbol{Y})=\mathbb{E}_{\boldsymbol{z}}\|\mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{Y}+\boldsymbol{z})-(\boldsymbol{y}-\boldsymbol{z})\|_2^2. \tag{6}$$

Similar as the image ER2R loss, the VER2R loss can also be proved as an unbiased estimator of its supervised counterpart. See the theorem below.

Theorem 3.2. *Consider the noisy video model $\boldsymbol{Y}=\boldsymbol{X}+\boldsymbol{N}$ with independent noise across frames and centering at the frame $\boldsymbol{y}=\boldsymbol{x}+\boldsymbol{n}$. Assume that conditioned on $\boldsymbol{X}$, $\boldsymbol{z}$ and $\boldsymbol{n}$ are i.i.d. and independent from noise in other frames. Then the VER2R loss function defined by* (6) *satisfies*

$$\mathbb{E}_{\boldsymbol{Y}}\ell_{VER2R}(\boldsymbol{\theta})=\mathbb{E}_{\boldsymbol{X},\boldsymbol{N},\boldsymbol{z}}\|\mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{Y}+\boldsymbol{z})-\boldsymbol{x}\|_2^2+const. \tag{7}$$

Proof. See the supplemental materials. □

Note that Theorem 3.2 holds true regardless of the architecture of $\mathcal{G}_{\boldsymbol{\theta}}$. Specifically, the optical flow, motion correction and temporal fusion process in our method does not violate the conditions in Theorem 3.2 . One limitation of Theorem 3.2 is it assumes the availability of prior information regarding $p(\boldsymbol{N}|\boldsymbol{X})$.

The entire NN is trained with the VER2R loss over all volumes $\{\boldsymbol{Y}_i\}$ to denoise every centering frame $\boldsymbol{y}_i$ in the video:

$$\mathcal{L}^{all}(\boldsymbol{\theta}) = \frac{1}{T}\sum_{i=1}^{T} \ell_{VER2R}(\boldsymbol{\theta}; \boldsymbol{Y}_i). \tag{8}$$

It is worth noting that the adjacent volumes $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_{i+1}$ are overlapped and correlated due to the sliding window used to generate the volumes. However, despite this correlation, $\mathcal{L}^{all}(\boldsymbol{\theta})$ can be regarded as an empirical approximation to the supervised loss $\mathbb{E}_{\boldsymbol{Y}}\ell_{VER2R}(\boldsymbol{\theta})$, as guaranteed by Theorem 3.2. This means that, in principle, the self-supervised training of the neural network using the VER2R loss can closely mimic supervised training, provided that the number of volumes T is sufficiently large.

Remark 3.2.1. *Many video denoisers have pointed out that flow estimation errors in challenging cases, such as occlusion or strong noise, can cause noticeable artifacts in the results. Our self-supervised training with VER2R loss implicitly addressed this issue to certain degree, as it mimics the supervised training with clean target video according to our theoretical analysis.*

Temporal attention

To exploit spatio-temporal redundancy, we have incorporated both a spatial denoising module and a temporal fusion module in our NN. The spatial denoising module is pre-trained using an unsupervised image ER2R loss to preprocess the noisy frames. The preprocessed frames are then aligned using optical flow estimated by DIS (Kroeger et al. 2016). Although this image alignment process can handle large motions between frames, some misalignment errors can occur in certain regions, which are handled by a residual NN. The residual NN takes the pre-aligned neighboring frames and the spatially denoised centering frame as input and outputs the neighboring frames with refined alignment.

At the last stage of temporal fusion, we fuse the well-aligned neighbouring frames and the spatially-denoised centering frame to obtain the final denoised centering frame. To achieve this, we sample pixels in the temporal neighborhood of each pixel in the centering frame and aggregate them using weights predicted by a lightweight NN. This is where the attention mechanism comes into play in the network.

Let $\widetilde{\boldsymbol{Y}}_i$ (i denotes the index of the centering reference frame) denote the intermediate frames that are passed into the temporal fusion module and $\mathcal{W}_{\boldsymbol{\theta}^w}$ denote the weight network for fusion. We use the notation $\widetilde{\boldsymbol{Y}}_i[j,k]$ to represent the j-th pixel at the k-th frame. For each reference pixel j at the centering frame, the weight network outputs weights for pixels in a temporal neighbourhood, denoted by $\left\{\mathcal{W}_{\boldsymbol{\theta}^w}(\widetilde{\boldsymbol{Y}}_i)[j,k]\right\}_{k=-t_0+i}^{t_0+i}$. Then the attention module produces an output as

$$\hat{\boldsymbol{x}}_i[j] = \sum_{k=-t_0+i}^{t_0+i} \mathcal{W}_{\boldsymbol{\theta}^w}(\widetilde{\boldsymbol{Y}}_i)[j,k] \cdot \widetilde{\boldsymbol{Y}}_i[j,k], \tag{9}$$

where $\hat{\boldsymbol{x}}_i[j]$ denotes the j-th pixel in $\hat{\boldsymbol{x}}_i$.

Network architecture. Both the spatial denoising module and the temporal attention module are based on the U-Net with skip connections, which contains two and three Downsampling(Upsampling) blocks respectively. The motion-correction module is a five-layer DnCNN(Zhang et al. 2017). See the supplemental materials for more details.

4 Experiments

The experiments are conducted on two tasks: additive white Gaussian noise(AWGN) removal and real raw video denoising. For all the experiments, we train our network directly on the test video itself.

Implementation detail

Our method is implemented using PyTorch. In the first stage of training, the iteration number does not exceed 1500 or 30N (the number of frames). In the second stage, we train the whole network for 50 epochs to denoise each frame sequentially. Recall that we use DIS optical flow (Kroeger et al., 2016) to warp the frames for large motion compensation after the spatial denoising block $\mathcal{S}_{\boldsymbol{\theta}^s}$. To speed up training, we only calculate the flow once using the initial value of $\widetilde{\boldsymbol{y}}_i$ and deactivate the backward gradient flow through the warping operators. The exponential moving average of the intermediate outputs is used as the final prediction. The code is available at https://github.com/huanzheng551803/VER2R.

For comparison, we cite the results directly from the literature whenever possible. Otherwise, we run the authors' code with an effort on parameter tuning. If the code is not available, we leave the corresponding results blank in the table.

Remark 4.0.1. *In quantitative comparison of all unsupervised learning methods, the best result is emphasized in* **Bold***; and the second best one is emphasized in* <u>*Underline*</u>*.*

Experiments on AWGN removal

AWGN removal is evaluated following FastDVDNet (Tassano, Delon, and Veit 2020), a benchmark for video denoising. We use two datasets, DAVIS(Khoreva, Rohrbach, and Schiele 2018) and Set8(Tassano, Delon, and Veit 2020). DAVIS has a training set and a test set for supervised methods, with the test set containing 30 videos. Set8 includes 4 color sequences from the *Derf's Test Media collection*[1] and 4 color sequences from a GoPro camera. AWGN with a standard deviation σ varying from 10 to 50 uniformly is added to the datasets. The compared methods are VBM4D(Maggioni et al. 2011), DVDNet(Tassano, Delon, and Veit 2019), FastDVDnet(Tassano, Delon, and Veit 2020), PaCNet(Vaksman, Elad, and Milanfar 2021), FloRNN(Li et al. 2022), RFR(Lee

[1] https://media.xiph.org/video/derf

Dataset	σ	Non-learning	Supervised learning				Unsupervised learning		
		V-BM4D	DVDNet	FastDVDNet	PaCNet	FloRNN	UDVD	RFR	Ours
DVAIS	10	37.86/4.02	38.45/4.21	39.01/3.26	39.97/3.03	40.16/2.68	35.65/9.57	39.31/3.32	**39.52/3.21**
	20	34.02/10.67	35.95/7.39	35.98/6.72	37.10/6.16	37.52/4.73	35.40/8.98	36.15/7.00	**36.49/6.41**
	30	31.74/21.64	34.27/11.92	34.22/11.48	35.07/10.72	35.89/7.51	34.19/12.52	34.28/12.40	**34.60/11.09**
	40	30.12/36.50	33.01/17.79	32.96/17.65	33.57/16.91	34.66/11.01	32.93/19.72	32.92/19.41	**33.29/17.40**
	50	28.85/54.75	31.99/25.08	31.99/25.24	32.39/24.78	33.67/15.28	31.93/28.82	31.86/28.47	**32.25/26.09**
Set8	10	35.99/3.87	35.92/4.06	36.46/2.90	37.06/2.77	37.57/2.60	34.44/7.01	36.77/2.97	**37.55/2.96**
	20	32.17/9.54	33.38/7.06	33.44/6.06	33.94/5.59	34.67/4.95	33.26/7.53	33.64/6.51	**34.34/6.18**
	30	29.99/14.82	31.69/11.71	31.69/10.55	32.05/9.93	32.97/8.22	31.83/11.30	31.82/11.80	**32.45/10.88**
	40	28.57/26.20	30.46/17.59	30.47/16.45	30.70/15.79	31.75/12.35	30.58/17.88	30.52/18.78	**31.09/16.93**
	50	27.31/47.62	29.47/25.43	29.53/23.63	29.66/23.09	30.80/17.91	29.62/26.38	29.50/27.75	**30.05/25.29**

Table 1: The PSNR(dB)/ST-RRED results for AWGN removal on Set8 and DAVIS. For PSNR (ST-RRED), larger (smaller) value is better.

ISO	1600	3200	6400	12800	25600	mean
RViDeNet	47.74/3.93	45.91/3.33	43.85/6.66	41.20/11.03	41.17/26.22	43.97/10.23
MaskDnGAN	47.83/3.79	45.89/3.29	43.83/6.98	41.15/13.44	41.09/26.91	43.90/10.88
FloRNN	48.81/3.41	47.05/2.72	45.09/4.83	42.63/7.24	42.19/18.96	45.15/7.43
UDVD	47.94/3.65	46.36/2.97	44.69/5.39	42.22/8.10	41.97/19.3	44.63/7.89
R2R	48.46/4.10	46.67/3.23	44.70/6.13	41.99/10.21	41.71/20.11	44.70/8.76
Ours	**49.14/3.59**	**47.51/2.90**	**45.61/5.30**	**43.03/7.80**	**42.91/17.41**	**45.64/7.40**

Table 2: Raw video denoising results in PSNR(dB)/ST-RRED($\times 10^{-3}$) obtained by different methods. The columns correspond to different ISO levels, where larger levels results in noisier data. For PSNR (ST-RRED), larger (smaller) value is better.

et al. 2021), and UDVD(Sheth et al. 2021). VBM4D is a non-learning method, while the other methods are supervised video denoising benchmarks. RFR is a fine-tuned method based on a pre-trained supervised model, and UDVD is an unsupervised method. Our proposed method trains an NN for each test video individually, without using external datasets.

Quantitative comparison. Table 1 compares all methods on DAVIS and Set8 for AWGN removal, in terms of PSNR and ST-RRED. PSNR is Peak Signal to Noise Ratio, a measure of image quality, while ST-RRED is Spatio-temporal Reduced Reference Entropic Differences, a video quality measure that evaluates temporal distortions. Our method outperformed the non-learning method VBM4D and two unsupervised methods UDVD and RFR by a large margin in almost all settings. Our method remained competitive with the top supervised method in terms of PSNR, and was comparable to one supervised method FastDVDNet on DAVIS in terms of ST-RRED. However, since motion compensation is more challenging when no external training data with clean videos is available to guide the process, our self-supervised method achieved less performance gain in terms of ST-RRED, especially on Set8, which contains 4 Go-Pro videos with vigorous background motion.

Experiments on real raw video dataset

The experiment uses the real raw video dataset (Yue et al. 2020b), comprising 11 videos captured at 5 ISO levels with a surveillance camera. Each video contains 7 frames, with 10 different noise realizations captured per frame and averaged to obtain an estimated clean version. The dataset is divided into a training set and a test set, with the first six video sequences forming the training set and the remaining five forming the test set. To ensure a fair comparison with UDVD, which trains a universal model for all test videos, we also trained our spatial denoiser $\mathcal{S}_{\boldsymbol{\theta}^s}$ on all test video sequences.

Note that our proposed method, ER2R, extends the R2R method from Gaussian white noise to general random noise. To evaluate the benefits of using ER2R over R2R in real noise, we compare the results obtained by replacing ER2R with R2R in our method. Following (Yue et al. 2020b), the real noise is modeled as a mixture of Poisson and Gaussian noise, with their mixture weights estimated by the authors. Specifically, given the noisy observation $\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{n}_i$, where $\boldsymbol{x}_i$ is the clean image and $\boldsymbol{n}_i$ is the real noise, we model $\boldsymbol{n}_i$ as $\boldsymbol{n}_i = \boldsymbol{n}_{i,P} + \boldsymbol{n}_{i,G}$, where $\boldsymbol{n}_{i,P} \sim \mathcal{P}(\alpha \boldsymbol{x}_i)/\alpha - \boldsymbol{x}_i$, $\boldsymbol{n}_{i,G} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, and α and σ are parameters related to noise variance. To generate an independent Poisson noise in our EVR2R, we sample from either $\mathcal{P}(\alpha \boldsymbol{y}_i)/\alpha - \boldsymbol{y}_i$ or $\mathcal{P}(\alpha \mathcal{S}_{\boldsymbol{\theta}^s}(\boldsymbol{y}_i))/\alpha - \mathcal{S}_{\boldsymbol{\theta}^s}(\boldsymbol{y}_i)$. When applying R2R to Poisson noise, we approximate it using Gaussian noise with the same variance.

Quantitative Comparison. Five methods are used for comparison: three supervised methods including FloRNN(Li et al. 2022), MaskDnGAN(Paliwal, Zeng, and Kalantari 2021), RViDeNet(Yue et al. 2020b) which are specifically designed for raw data, one unsupervised method UDVD (Sheth et al. 2021) and R2R (Pang et al. 2021) with the same pipeline as ours. The other compared methods in

Figure 2: Real raw denoising on CVPD dataset. All raw data has been transferred to RGB domain for visualization. Our method requires no external clean videos for training opposite to supervised methods..

the experiments of AWGN removal are not included here as either the model is not applicable to raw data, or the code for raw data is not released. See Table 2 for the quantitative comparison and Figure 2 for some visual results. The proposed method outperforms all the compared methods by a large margin, despite that it requires no external clean videos for training opposite to supervised methods. The experimental results also demonstrated the superiority of our proposed loss ER2R over R2R when dealing with real noise.

More discussion

Ablation studies. To evaluate the effectiveness and efficiency of each component of our method, we conduct the following experiments on DAVIS for AWGN removal with noise level $\sigma = 20, 40$: (a) only using the spatial denoiser; (b) w/o the motion correction module; (c) w/o the temporal attention module; (d) using 3D spatial-temporal kernels for fusion instead of only fusing pixels along the temporal dimension in ours; (e) using 3 adjacent frames instead of 5 frames used in ours. See Table 3 for the results. The first three studies show the effectiveness of each module in our method, and the fourth study implies that motion compensation is sufficient and a fusion along only the temporal dimension is efficient to obtain a well denoised centering frame. The last study shows that our method exploits the temporal redundancy extensively in a large neighbourhood.

Studies	$\sigma = 20$	$\sigma = 40$
using only spatial denoiser	34.77	31.91
w/o motion correction	36.17	33.18
w/o temporal attention	35.36	32.70
using 3D attention kernels	36.37	**33.30**
using 3 adjacent frames	36.09	33.04
ours	**36.49**	33.29

Table 3: Comparison of PSNR (dB) under different studies.

Computational efficiency. We compare the running time of several methods on removing AWGN from a RGB video of 85 frames and spatial size 548×960. See Table 4 for the results. UDVD(S) indicates it is trained on the single test video. All the deep learning methods are conducted using the same computing infrastructure as ours for time counting. It can be seen that the proposed method is much faster than VBM4D(Maggioni et al. 2011), MF2F(Dewil et al. 2021) and UDVD-S(Sheth et al. 2021). In addition, the supervised method PaCNet is also slow due to its call of the time-consuming patch matching during inference. RFR(Lee et al. 2021) is the fastest among all, as it used a pre-trained video denoiser which accelerates the network training. Overall, our method is still competitive in terms of computational efficiency.

Category	Methods	Stage	Time (s)	PSNR
Tradional	VBM4D	*	2777	34.27
Supervised	PaCNet	I	2890	35.07
Self-supervised	UDVD(S)	T+I	127800	33.68
	MF2F	T+I	4950	33.91
	RFR	T+I	1326	34.28
	Ours	T+I	1605	34.60

Table 4: Running time(second) of processing a RGB video sequence of 85 frames and spatial size 540×960; Stage 'I'('T') indicate the time is counted for the inference(training) stage.

Theoretical limitation. The proposed unsupervised denoising method shows good empirical performance on general measurement noise. However, the mathematical guarantee of the proposed loss requires prior knowledge of the noise distribution, which may not always be available in practice. In the future, we plan to investigate self-supervised losses with mathematical guarantees that do not require prior knowledge of the measurement noise distribution.

5 Conclusion

In this paper, we introduced an unsupervised deep video denoiser that leverages a self-supervised VER2R loss and a spatial-temporal denoising neural network equipped with a lightweight temporal attention module. Experimental results show that our method outperforms existing unsupervised methods by a noticeable margin and is competitive with state-of-the-art supervised methods, all without requiring any external training samples.

Acknowledgment

The authors would like to thank the support by Singapore MOE Academic Research Fund (AcRF) with WBS number R-146-000-315-114.

References

Aharon, M.; Elad, M.; and Bruckstein, A. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11): 4311–4322.

Arias, P.; and Morel, J.-M. 2018. Video Denoising via Empirical Bayesian Estimation of Space-Time Patches. *Journal of Mathematical Imaging and Vision*, 60(1): 70–93.

Batson, J.; and Royer, L. 2019. Noise2Self: Blind Denoising by Self-supervision. *Proc. ICML*.

Cai, J.-F.; Ji, H.; Shen, Z.; and Ye, G.-B. 2014. Data-driven tight frame construction and image denoising. *Applied and Computational Harmonic Analysis*, 37(1): 89–105.

Chan, K. C.; Wang, X.; Yu, K.; Dong, C.; and Loy, C. C. 2021. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proc. CVPR*, 4947–4956.

Dabov, K.; Foi, A.; Katkovnik, V.; and Egiazarian, K. O. 2007. Color Image Denoising via Sparse 3D Collaborative Filtering with Grouping Constraint in Luminance-Chrominance Space. In *Proc. ICIP*, 313–316.

Davy, A.; Ehret, T.; Morel, J.-M.; Arias, P.; and Facciolo, G. 2019. A Non-Local CNN for Video Denoising. In *Proc ICIP*, 2409–2413. IEEE.

Dewil, V.; Anger, J.; Davy, A.; Ehret, T.; Facciolo, G.; and Arias, P. 2021. Self-Supervised Training for Blind Multi-frame Video Denoising. In *Proc. WCACV*, 2724–2734.

Ehret, T.; Davy, A.; Morel, J.-M.; Facciolo, G.; and Arias, P. 2019. Model-Blind Video Denoising via Frame-to-frame Training. In *Proc. CVPR*, 11369–11378.

Khoreva, A.; Rohrbach, A.; and Schiele, B. 2018. Video Object Segmentation with Language Referring Expressions. In *Proc. ACCV*, 123–141. Springer.

Kroeger, T.; Timofte, R.; Dai, D.; and Van Gool, L. 2016. Fast Optical Flow Using Dense Inverse Search. In *Proc. ECCV*, 471–488. Springer.

Krull, A.; Buchholz, T.-O.; and Jug, F. 2019. Noise2Void-Learning Denoising from Single Noisy Images. In *Proc. CVPR*, 2129–2137.

Laine, S.; Karras, T.; Lehtinen, J.; and Aila, T. 2019. High-Quality Self-Supervised Deep Image Denoising. *Proc. NeurIPS*, 32.

Lee, S.; Cho, D.; Kim, J.; and Kim, T. H. 2021. Restore from Restored: Video Restoration with Pseudo Clean Video. In *Proc. CVPR*, 3537–3546.

Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; and Aila, T. 2018. Noise2Noise: Learning Image Restoration without Clean Data. In *Proc. ICML*.

Li, J.; Wu, X.; Niu, Z.; and Zuo, W. 2022. Unidirectional Video Denoising by Mimicking Backward Recurrent Modules with Look-Ahead Forward Ones. In *Proc. ECCV*, 592–609. Springer.

Maggioni, M.; Boracchi, G.; Foi, A.; and Egiazarian, K. 2011. Video Denoising Using Separable 4D Nonlocal Spatiotemporal Transforms. In *Image Processing: Algorithms and Systems IX*, volume 7870, 787003. International Society for Optics and Photonics.

Maggioni, M.; Huang, Y.; Li, C.; Xiao, S.; Fu, Z.; and Song, F. 2021. Efficient Multi-Stage Video Denoising with Recurrent Spatio-Temporal Fusion. In *Proc. CVPR*, 3466–3475.

Mildenhall, B.; Barron, J. T.; Chen, J.; Sharlet, D.; Ng, R.; and Carroll, R. 2018. Burst Denoising with Kernel Prediction Networks. In *Proc. CVPR*, 2502–2510.

Paliwal, A.; Zeng, L.; and Kalantari, N. K. 2021. Multi-stage raw video denoising with adversarial loss and gradient mask. In *Proc ICCP*, 1–10. IEEE.

Pang, T.; Zheng, H.; Quan, Y.; and Ji, H. 2021. Recorrupted-to-Recorrupted: Unsupervised Deep Learning for Image Denoising. In *Proc. CVPR*, 2043–2052.

Pérez, J. S.; Meinhardt-Llopis, E.; and Facciolo, G. 2013. TV-L1 Optical Flow Estimation. *Image Processing On Line*, 2013: 137–150.

Quan, Y.; Chen, M.; Pang, T.; and Ji, H. 2020. Self2Self with Dropout: Learning Self-Supervised Denoising from Single Image. In *Proc. CVPR*, 1890–1898.

Rudin, L. I.; Osher, S.; and Fatemi, E. 1992. Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D: nonlinear phenomena*, 60(1-4): 259–268.

Sheth, D. Y.; Mohan, S.; Vincent, J. L.; Manzorro, R.; Crozier, P. A.; Khapra, M. M.; Simoncelli, E. P.; and Fernandez-Granda, C. 2021. Unsupervised Deep Video Denoising. In *Proc. CVPR*, 1759–1768.

Tassano, M.; Delon, J.; and Veit, T. 2019. Dvdnet: A Fast Network for Deep Video Denoising. In *Proc. ICIP*, 1805–1809. IEEE.

Tassano, M.; Delon, J.; and Veit, T. 2020. Fastdvdnet: Towards Real-time Deep Video Denoising without Flow Estimation. In *Proc. CVPR*, 1354–1363.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2018. Deep Image Prior. In *Proc. CVPR*, 9446–9454.

Vaksman, G.; Elad, M.; and Milanfar, P. 2021. Patch Craft: Video Denoising by Deep Modeling and Patch Matching. In *Proc. ICCV*, 2157–2166.

Xia, Z.; Perazzi, F.; Gharbi, M.; Sunkavalli, K.; and Chakrabarti, A. 2020. Basis Prediction Networks for Effective Burst Denoising with Large Kernels. In *Proc. CVPR*, 11844–11853.

Xu, X.; Li, M.; Sun, W.; and Yang, M.-H. 2020. Learning Spatial and Spatio-Temporal Pixel Aggregations for Image and Video Denoising. *IEEE Trans. Image Process*, 29: 7153–7165.

Yang, X.; Xu, Y.; Quan, Y.; and Ji, H. 2020. Image denoising via sequential ensemble learning. *IEEE Transactions on Image Processing*, 29: 5038–5049.

Yu, S.; Park, B.; Park, J.; and Jeong, J. 2020. Joint Learning of Blind Video Denoising and Optical Flow Estimation. In *Proc. CVPR Workshops*.

Yue, H.; Cao, C.; Liao, L.; Chu, R.; and Yang, J. 2020a. Supervised Raw Video Denoising with a Benchmark Dataset on Dynamic Scenes. In *Proc. CVPR*, 2301–2310.

Yue, H.; Cao, C.; Liao, L.; Chu, R.; and Yang, J. 2020b. Supervised Raw Video Denoising with a Benchmark Dataset on Dynamic Scenes. In *Proc. CVPR*, 2301–2310.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process*, 26(7): 3142–3155.