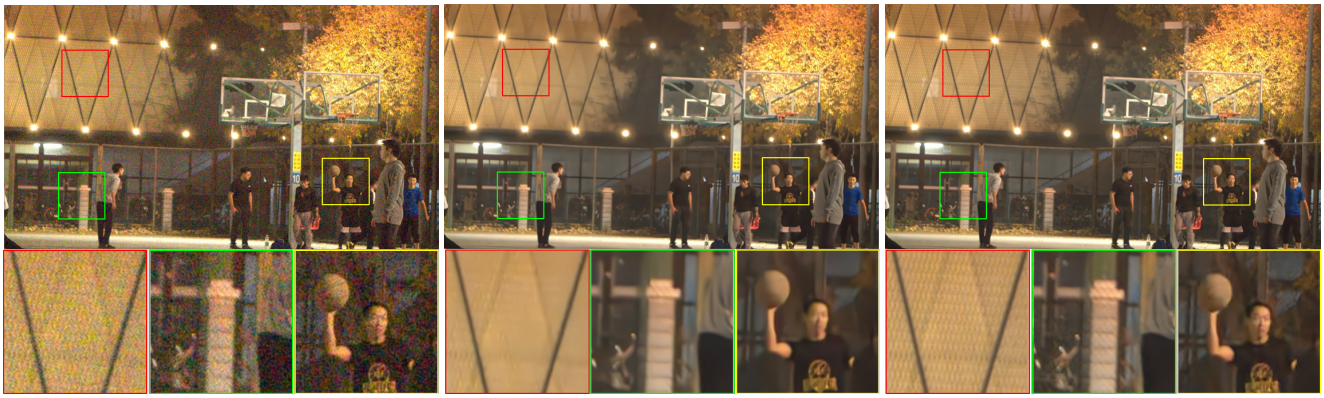
Efficient Multi-Stage Video Denoising with Recurrent Spatio-Temporal Fusion

Matteo Maggioni*, Yibin Huang*, Cheng Li*, Shuai Xiao, Zhongqian Fu, Fenglong Song
Huawei Noah's Ark Lab
{matteo.maggioni, huangyibin1, licheng89, xiaoshuai7, fuzhongqian, songfenglong}@huawei.com



Noisy input – ISO 25600. RViDeNet [42] – 1965.04 GFLOPs. Ours – 5.38 GFLOPs.

Figure 1: Our method shows better noise suppression and detail preservation than the state-of-the-art RViDeNet at a much lower complexity. The input is a 1080p image from the CRVD dataset [42] captured with a IMX385 sensor.

Abstract

In recent years, denoising methods based on deep learning have achieved unparalleled performance at the cost of large computational complexity. In this work, we propose an Efficient Multi-stage Video Denoising algorithm, called EMVD, to drastically reduce the complexity while maintaining or even improving the performance. First, a fusion stage reduces the noise through a recursive combination of all past frames in the video. Then, a denoising stage removes the noise in the fused frame. Finally, a refinement stage restores the missing high frequency in the denoised frame. All stages operate on a transform-domain representation obtained by learnable and invertible linear operators which simultaneously increase accuracy and decrease complexity of the model. A single loss on the final output is sufficient for successful convergence, hence making EMVD easy to train. Experiments on real raw data demonstrate that EMVD outperforms the state of the art when complexity is constrained, and even remains competitive against methods whose complexities are several orders of magnitude higher. Further, the low complexity and memory requirements of EMVD enable real-time video denoising on commercial SoC in mobile devices.

*Authors contributed equally.

1. Introduction

Even with the advance of technology, digital images are invariably affected by several inherent or external disturbing factors due to the stochastic nature of the image formation processes (e.g., photon counting) [16], use of compact camera hardware (e.g., mobile sensors or lenses) [42], and/or challenging acquisition settings (e.g., low light). Because of this, a number of methods (i.e., an image processing pipeline) must be applied to the low-quality observed data to generate a final high-quality output image. Denoising is particularly important because it is typically at the beginning of the pipeline, and thus its output has a direct effect on all other operations [39].

In the past decades, a plethora of image denoising algorithms have been proposed in the literature [5, 12], but the current state of the art is dominated by deep learning methods based on convolutional neural networks (CNNs) [43, 25, 27]. Video denoising models exploits the temporal correlation inherently present in natural videos and thus achieve better performance than single-frame methods [28, 1, 30, 36, 42], however their computational requirements make real-time processing unattainable on most hardware unless some compromise in image quality is made [15].

In this work we propose EMVD, an Efficient Multi-stage Video Denoising method to drastically reduce the complex-

ity required to achieve high-quality results. Firstly, noise in the input frame is reduced by recursively fusing all past frames in the video. Then, a denoising stage removes any remaining noise in the fused image. Finally, a refinement stage is applied to the denoised image to further improve its quality by adaptively restoring the high-frequency details. All stages are performed within a domain generated by learnable and invertible linear transform operators that jointly decorrelate color and frequency information. As can be seen in Fig. 1, the proposed EMVD is able to outperform more complex state-of-the-art methods [42] at a fraction of the computational cost. In summary, the main contributions of this work are

- **High-Quality Efficient Denoising.** The proposed EMVD leverages spatio-temporal correlation of natural videos through specialized processing stages, namely temporal fusion, spatial denoising, and spatio-temporal refinement. This design allows to significantly reduce the model complexity without compromising its denoising capabilities.
- **Interpretable Design.** All stages in the proposed EMVD have a clear objective and will naturally converge to the desired behavior without any explicit supervision. As a result, the inner workings of the proposed EMVD can be easily inspected and controlled at both inference and training time.
- **Learnable and Invertible Transforms.** Linear transform operators, implemented as learnable convolutional layers, are used to optimally decorrelate color and frequency information. The learnable parameters are regularized in the loss to ensure transform invertibility. This simultaneously allows to reduce complexity and increase accuracy of the model.

2. Related works

Image Denoising. Denoising is a long-studied research topic [5, 12], however the numerous works proposed in the recent past [43, 25, 27, 8] suggest that the interest towards this problem is still very active, especially in the more challenging case of real raw data [3, 20, 23, 9].

Classical methods heavily exploit nonlocal image priors [4, 12] and still provide outstanding performance today. Despite being designed for Gaussian noise, such methods can be also applied to real raw data when a variance-stabilizing transformation (VST) is used [29]. More recently –and since the pioneering work [14]– deep learning and CNNs have become the *de-facto* standard solution for all vision problems, including denoising. CNN-based methods most notably leverage residual learning [43], wavelet decomposition [27], attention mechanisms [25], and spatially adaptive processing [34, 8].

Video Denoising. Natural videos exhibit a strong correlation along the temporal dimension, i.e., pixels at corresponding locations in consecutive frames are likely to be very similar. One viable strategy to account for temporal correlation is to explicitly estimate the motion in the video with, e.g., block matching [28], optical flow [7, 32], kernel-prediction networks [30], or deformable convolutions [37, 38]. With that, frames can be aligned before filtering to aid the restoration task. However, since motion estimation is a challenging and computationally demanding problem, a different line of research suggests to implicitly deal with motion through, e.g., spatio-temporal attention modules which recursively aggregate features at different time steps [22, 41, 38, 13]. These methods can be further categorized into multi-frame approaches, whose inputs include several consecutive frames that are jointly processed by the model [28, 11, 35, 36, 42], or recurrent approaches where images [32, 15] and/or features [19, 17] obtained from the previous time step are used as additional input to estimate the current frame. Several models have been designed with efficiency in mind, yet real-time computation is still unattainable on most hardware [17, 36], or performance is not on-par with the state of the art [15].

3. Method

3.1. Observation Model

The goal of our denoising algorithm is to obtain an estimate of the clean video from the observed noisy data at a very low complexity. The observation model is defined as

$$z_t(x) = y_t(x) + \eta_t(x), \tag{1}$$

where $t \in T \subset \mathbb{N}$ is the temporal index of the frame in the video, $x \in X \subset \mathbb{N}^2$ is a spatial pixel position in the frame, $z \in \mathbb{R}^{H \times W \times C}$ is the observed noisy raw video in packed form [18] having $H \times W$ resolution and $C = 4$ color channels (e.g., RG_1G_2B), y is the underlying (unknown) noise-free data to be estimated, and $\eta_t \sim \mathcal{N}\left(0, \sigma_t^2(y_t)\right)$ is a noise realization drawn from a heteroskedastic Gaussian distribution with signal-dependent variance

$$\sigma_t^2(y_t) = a_t y_t + b_t \tag{2}$$

modeling signal-dependent (shot) and signal-independent (read) noise sources parametrized by a_t and $b_t \in \mathbb{R}$, respectively. Since many robust estimation methods exist [16, 2], we assume such noise parameters to be known for the given sensor and camera ISO.

3.2. Learnable Invertible Transforms

The proposed method employs learnable transform operators –inspired by YUV and wavelet transforms– to decorrelate color and frequency information of the raw data.

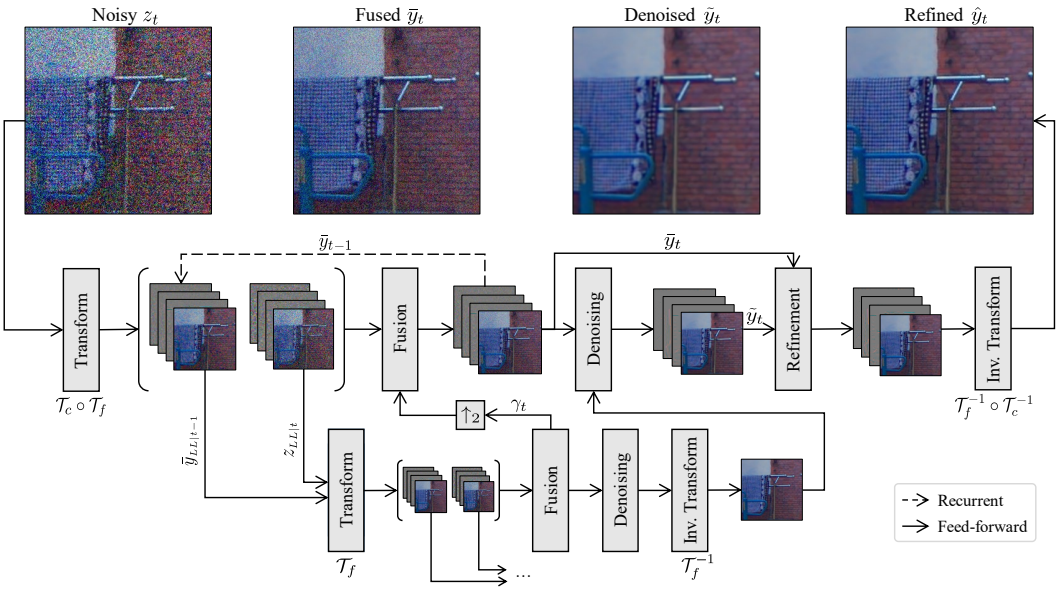


Figure 2: Architecture of the proposed multi-stage video denoising EMVD. Refer to Section 3 for details.

These operators are linear and designed to be invertible, and thus can be implemented as standard convolutional operations with deliberately regularized weights.

Color Transform. The color transform $\mathcal{T}_c$ is implemented as a point-wise convolution whose kernel is constrained to be an orthonormal matrix $M \in \mathbb{R}^{C\times C}$ which decorrelates the $C = 4$ colors (e.g., RG_1G_2B) in the CFA image to a luminance-chrominance representation [6]. Being the color transform orthonormal, its inverse $\mathcal{T}_c^{-1}$ is simply implemented as another point-wise convolution with kernel initialized as $M' = M^\top$. In this work, the weights in the forward and inverse matrices are not shared to allow more degrees of freedom. The resulting transform is therefore biorthogonal and its invertibility is enforced by a loss term defined as

$$\mathcal{L}_c = \left|\left|M \cdot M' - I_C\right|\right|_F^2, \tag{3}$$

where $\cdot$ is matrix multiplication, and I_C is the identity matrix of rank C, and $\|\cdot\|_F$ denotes the Frobenius norm.

Frequency Transform. Inspired by biorthogonal wavelets, we design a transform $\mathcal{T}_f : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H/2\times W/2\times 4C}$ to decorrelate the input frequencies into four half-resolution components, namely the low-pass LL and high-pass $\{LH, HL, HH\}$ subbands. The transform is again linear and thus can be implemented as a strided convolution with four $n \times n$ kernels initialized as the outer product of some chosen wavelet decomposition filters $\psi \in \mathbb{R}^{2\times n}$, being $n \in \mathbb{N}^+$ the (even) length of the wavelet filters (e.g., $n = 2$ for Haar). Conversely, the inverse operator $\mathcal{T}_f^{-1}$ is implemented as a transposed convolution with kernels initialized this time from the corresponding reconstruction filters $\phi \in \mathbb{R}^{2\times n}$. Note that the same $\mathcal{T}_f$ can be recursively applied on the LL subband to produce a multi-scale decomposition of the input.

We enforce invertibility of the transform by adding a loss term on the matrix form of the filters as

$$\mathcal{L}_f = \left|\left|\psi \cdot \phi^\top - I_2\right|\right|_F^2, \tag{4}$$

where I_2 is the identity matrix of rank 2. As such, the proposed method is learning the 1-D filter representation of the frequency transform, and not the convolutional form of the kernel. A different strategy would be forcing the learned filters to follow a wavelet parametric model [40]. Thus our approach might generate filters that do not satisfy basic wavelet properties, however our filters have more degrees of freedom and still produce an invertible transformation.

Note that the composition of the color and frequency transform $\mathcal{T}_c \circ \mathcal{T}_f$ is still linear and invertible. A joint application of the two transforms allows to simultaneously increase accuracy and reduce complexity of the model because the energy of the meaningful part of the image is decorrelated from the energy of the noise and, at the same time, spatial resolution of the data is halved. A diagram of the proposed EMVD is illustrated in Fig. 2. In the remainder of this paper, we will assume the data to be given already in the transform domain, thus, for the sake of notation simplicity, we will omit the transform operators.

3.3. Fusion Stage

The first processing stage of the proposed EMVD is temporal fusion. The objective of this stage is to maximally reduce the noise present in the image using the temporal cor-

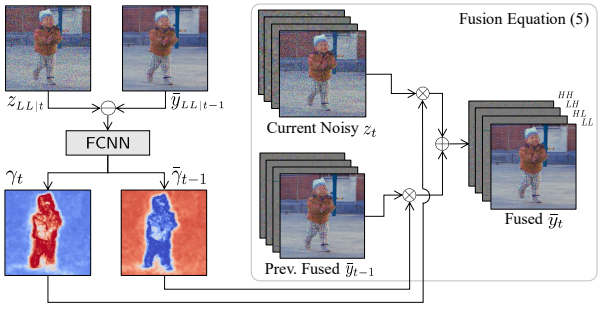


Figure 3: The predicted weights γ_t discriminate the dynamic (red) and static (blue) regions between consecutive frames to minimize the noise in the output fused image.

relation inherently present in the video without introducing any temporal artifact or degrading image structures. Formally, fusion is defined as a recursive convex combination

$$\bar{y}_t(x) = \bar{y}_{t-1}(x)\bar{\gamma}_{t-1}(x) + z_t(x)\gamma_t(x), \tag{5}$$

where z_t is the transformed noisy frame at time t, $\bar{y}_{t-1}$ is the transformed fused output frame at previous time $t-1$, and $\gamma \in \mathbb{R}^{H/2\times W/2\times 1}$ are non-negative convex weights satisfying the condition $\bar{\gamma}_{t-1}(x) + \gamma_t(x) = 1$ at any given transform-domain location $x \in \chi \subset \mathbb{N}^{H/2\times W/2\times 4C}$. The initial condition for (5) is $\bar{y}_0 \equiv z_0$, as no previous frame is available at time $t = 0$. Note that the number of channels of γ is 1, so fusion of the full $4C$ input channels is achieved by element-wise broadcasting. As a result, we apply the same weights to each channel (i.e., subband) of the input frames. Different weights could be predicted for different input channels, but this would significantly increase the difficulty of the prediction task, as well as the memory requirements.

The weights in (5) are predicted by a fusion network, which we call FCNN, defined as follows:

$$\{\gamma_t, \bar{\gamma}_{t-1}\} = \mathsf{FCNN}\Big(\,|z_{LL|t} - \bar{y}_{LL|t-1}|\,,\ \hat{\sigma}_t^2\Big), \tag{6}$$

where $|\cdot|$ denotes absolute value, $z_{LL|t}$ is the low-pass of the transformed noisy input frame, $\bar{y}_{LL|t-1}$ is the low-pass of the previous fused frame, and $\hat{\sigma}_t^2 = \sigma_t^2\left(z_{LL|t}\right)$ is the variance of the input frame computed as in (2). Note that the variance is approximated using the low-pass of z_t as proxy for the (unknown) noise-free y_t. In order to maintain convexity of (5), the output layer of FCNN is activated by a sigmoid function. Fig. 3 illustrates the diagram of the fusion network; note how the predicted weights γ_t clearly separate the dynamic regions from the static ones. As a result, the output fused image is generated by adaptively averaging the incoming frames. Note that fusion is performed at a lower image resolutions, a single position at any given scale i actually corresponds to a $2^i \times 2^i$ neighborhood in the original image, hence allowing some degree of motion compensation. As shown in Fig. 2, when multiple scales are available, the weights obtained from the fusion stage at the lower scale are upsampled and concatenated to the input of (6) to provide additional guidance information.

Finally, (6) can be interpreted as a special case of kernel-predicting network [30] with 1×1 kernels, thus (5) can be trivially extended to a general (convolutional) form as

$$\bar{y}_t(x) = \bar{y}_{t-1}(x) \circledast \bar{k}_{t-1}(x) + z_t(x) \circledast k(x), \tag{7}$$

where $\circledast$ denotes convolution, k is the spatially adaptive kernel of size $p \times p$ applied to the noisy frame, and $\bar{k}_{t-1}$ is the kernel of size $\bar{p} \times \bar{p}$ applied to the previous one. Practically, the kernels in (7) can be obtained by letting the output layer of the fusion network predict $\bar{p}^2 + p^2$ channels activated by a softmax function to ensure that the fusion equation is still convex.

3.4. Denoising Stage

Noise in the output fused image $\bar{y}_t$ is reduced by (5) but not completely. This inevitably occurs when, e.g., motion cannot be compensated effectively or when the amount of processed temporal data is not enough to increase the signal-to-noise ratio (SNR) in the frame. Thus, we use a denoising network, called DCNN, to remove any remaining noise in $\bar{y}_t$ as

$$\tilde{y}_t = \mathsf{DCNN}\Big(\bar{y}_t,\ z_{LL|t},\ \bar{\sigma}_t^2\Big), \tag{8}$$

where $\tilde{y}_t$ is the denoised image, and $\bar{\sigma}_t^2$ is the noise variance of the fused image $\bar{y}_t$. The input also includes the low-pass of the current noisy frame $z_{LL|t}$ so that the network has the chance to extract valuable information from the unadulterated noisy input. When multiple scales are available, the image estimated at the lower scale is concatenated to the input of (8).

Denoising the fused image $\bar{y}_t$ is easier than directly denoising the input z_t, but the form of the variance $\bar{\sigma}_t^2$ is highly complex as it depends on the signal-dependent variance at frame t as well as on the cumulative effect fusing all previous frames $t \in \{0, \ldots, t-1\}$. Nevertheless, fusion itself is linear, so we are able to define a recursive formulation of the fused variance using basic statistical properties[1] by expanding (5) into (2) as

$$\begin{aligned}\bar{\sigma}_t^2 &\equiv \sigma_t^2\left(\bar{y}_{LL|t}\right) \\ &= \bar{\gamma}_{t-1}^2\sigma_{t-1}^2\left(\bar{y}_{LL|t-1}\right) + \gamma_t^2\sigma_t^2\left(z_{LL|t}\right),\end{aligned} \tag{9}$$

where the initial condition is $\sigma_t^2\left(\bar{y}_{LL|0}\right) \equiv \sigma_t^2\left(z_{LL|0}\right)$, and the covariance term is zero as we assume that the noise is temporally independent. Note that, since $\gamma_t(x) \leq 1$ for all

[1] $\mathbb{V}\mathrm{ar}[aX + bY] = a^2\mathbb{V}\mathrm{ar}[X] + b^2\mathbb{V}\mathrm{ar}[Y] + 2ab\mathbb{C}\mathrm{ov}[X, Y]$

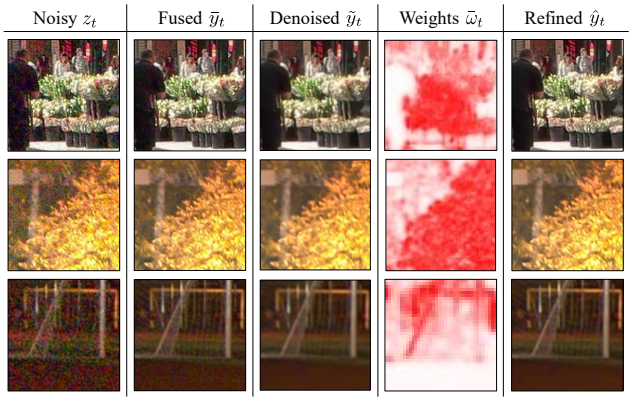


Figure 4: The red regions in the weights ω_t identify the high-frequency (e.g., edges and textures) information of the fused image used to refine the denoised image.

x and t, the variance (9) is (non-strictly) decreasing with time, thereby justifying the intuitive idea that fusion will progressively reduce the noise in the input.

3.5. Refinement Stage

Any denoising method is likely to introduce artifacts and loss of image details, especially when SNR of the input image is poor or when the complexity of model is significantly constrained. Thus, we propose to use a final refinement stage to combine the detailed –but still noisy– fused image $\bar{y}_t$ with the noise-free –but likely oversmoothed– denoised image $\tilde{y}_t$. In doing so, we expect to restore the fine details and textures potentially removed by the denoising, a task which is facilitated by our learned frequency representation which naturally decorrelates low- to high-frequency information. Formally, we seek to solve a refinement equation

$$\hat{y}_t(x) = \bar{y}_t(x)\bar{\omega}_t(x) + \tilde{y}_t(x)\tilde{\omega}_t(x), \tag{10}$$

with convex weights satisfying $\bar{\omega}_t(x) + \tilde{\omega}_t(x) = 1$ for all $x \in \chi$. The refinement weights are predicted by yet another network, called RCNN, operating as

$$\{\tilde{\omega}_t(x), \bar{\omega}_t\} = \mathsf{RCNN}\Big(\tilde{y}_t,\ \bar{y}_t,\ \bar{\sigma}_t^2\Big), \tag{11}$$

where convexity of the output weights is again ensured by applying a sigmoid activation on the final output layer. Although the formulation (10) is equivalent to the fusion equation (5), the refinement weights ω have a markedly different meaning than the fusion weights γ. In fact, as shown in Fig. 4, refinement weights are used to identify high-frequency information from the fused image whereas fusion weights are used to recursively aggregate consistent temporal information across consecutive frames to reduce the effect of the noise. Interestingly, no explicit supervision is required to converge to this behavior.

As shown in Fig. 2, the refinement network is only used at the higher scale of the frequency decomposition, even when lower scales are available. Finally, we highlight that (11) can be extended to predict kernels of arbitrary size analogously to [30].

4. Experiments

We compare the proposed EMVD against various state-of-the-art video denoising methods, namely VBM4D [28], RViDeNet [42], FastDVDnet [36], and EDVR [38]. In our experiments, we show performance of the aforementioned models at various levels of complexity, measured as floating-point operations (FLOPs). Additional technical details and experiments can be found in the supplementary materials.

To evaluate our model, we use the video benchmark dataset proposed in [42]. This consists of a real raw video dataset (CRVD) captured by a SONY IMX385 sensor and a synthesized dataset (SRVD) generated from [10]; all videos have five different ISO levels ranging from 1600 to 25600. Following [42], we use SRVD videos plus the scenes 1–6 from CRVD for training, hence keeping CRVD scenes 7–11 for objective validation. CRVD also includes few outdoor noisy raw videos without ground-truth which we use as test set to subjectively asses visual quality.

4.1. Training

We use training sequences composed of n patches of size 128×128 cropped at random spatio-temporal locations of the training videos, minding to preserve the CFA Bayer pattern [26]. Specifically, we use $n = 3$ for RViDeNet, $n = 5$ for FastDVDnet and EDVR, and $n = 25$ for EMVD. Note that EMVD is a recurrent models, and thus benefit from large values of n since the models are temporally unrolled during training to allow backpropagation through time. Differently, RViDeNet, FastDVDnet, and EDVR are multi-frame methods and thus n is equal to the number of frames used in their inputs.

The loss is defined as $\mathcal{L} = \mathcal{L}_r + \mathcal{L}_c + \mathcal{L}_f$, where $\mathcal{L}_r = \frac{1}{n}\sum_{t=1}^{n} \parallel \hat{y}_t - y_t \parallel_1$ denotes the mean L1 norm of the difference between each predicted $\hat{y}_t$ and ground-truth y_t frame in the sequence, and $\mathcal{L}_c$ and $\mathcal{L}_f$ are terms constraining invertibility of the color (3) and frequency (4) transforms, respectively. We train the networks using Adam optimizer [24] with batch size 16 and initial learning rate 1e-4. We apply a piece-wise constant decay which reduces the learning rate by a factor of 10 every 100000 iterations. All models are trained for an initial 300000 iterations on the CRVD and SRVD dataset, and then fine-tuned for an additional 300000 iterations on CRVD only. We implemented the proposed EMVD with Huawei MindSpore [31] and TensorFlow; both implementations show comparable accuracy and efficiency.

GFLOPs	Recurrent	$z_{LL\|t}$	$\mathcal{T}_c$	$\mathcal{T}_f$	RCNN	σ^2	PSNR / SSIM
5.38	$\bar{y}_{t-1}$	✓	✓	✓	✓	✓	**42.63 / 0.9851**
5.38	$\hat{y}_{t-1}$	✓	✓	✓	✓	✓	42.38 / 0.9840
5.12	$\bar{y}_{t-1}$	×	✓	✓	✓	✓	41.87 / 0.9831
5.31	$\bar{y}_{t-1}$	✓	×	✓	✓	✓	42.36 / 0.9839
5.38	$\bar{y}_{t-1}$	✓	✓	×	✓	✓	42.35 / 0.9838
5.94	$\bar{y}_{t-1}$	✓	✓	✓	×	✓	42.46 / 0.9848
5.18	$\bar{y}_{t-1}$	✓	✓	✓	✓	×	41.39 / 0.9795
5.42	$\hat{y}_{t-1}$	×	×	×	×	✓	41.35 / 0.9737

(a) Network structure.

GFLOPs	FCNN	DCNN	RCNN	PSNR / SSIM
2542.86	4 / 64	4 / 512	4 / 64	**44.51 / 0.9897**
1105.65	4 / 64	6 / 256	4 / 64	44.48 / 0.9895
86.23	3 / 64	3 / 64	3 / 64	43.57 / 0.9881
82.06	4 / 16	5 / 64	3 / 64	43.83 / 0.9883
79.52	4 / 16	6 / 64	1 / 32	44.05 / 0.9890
25.31	4 / 16	5 / 32	3 / 32	43.19 / 0.9869
9.85	4 / 16	5 / 16	3 / 16	42.73 / 0.9854
5.38	2 / 16	2 / 16	2 / 16	42.63 / 0.9851

(b) Number of convolutions / number of filters.

Table 1: Ablation study of the proposed EMVD evaluated on the raw CRVD dataset [42].

For RViDeNet we directly utilize the model and weights provided by the authors, and we train versions with reduced complexity using the same three-stage procedure suggested in the original paper [42]. For VBM4D we perform a grid search on the target sRGB image to find the optimal parameters that maximize the validation PSNR.

4.2. Ablation

Baseline. There are three CNNs involved in EMVD. Any backbone could be used, but, for the sake of efficiency, in all cases we use two convolutional layers (3×3 kernels, 16 filters) followed by ReLU activation plus one final output convolution. The inputs (i.e., images and variance) are always concatenated before processing. The output layers of both fusion and refinement CNNs are activated by a sigmoid, but no activation is applied to the denoising CNN. We use three decomposition scales. The frequency transform is initialized with Haar kernels, and the color transform is initialized as in [6]. This configuration is highlighted in yellow in all tables and correspond to 5.38 GFLOPs.

Network Structure. Table 1a reports an ablation study on the structure of proposed method. In the fusion (6), if we use the previous final output $\hat{y}$ instead of the previous fused image $\bar{y}$ PSNR decreases by 0.25dB; instead if we remove the low-pass noisy input $z_{LL|t}$ from the denoising (8) PSNR decreases by 0.76dB. The color transform, while only costing 0.08 GFLOPs, provides a sizable 0.27dB boost, and if we replace the learnable frequency transform with pixel shuffling [33] the PSNR reduces by 0.28dB. Then, if we remove the refinement stage, while increasing capacity of the denoising network as not to change overall complexity, we observe a 0.17dB decrease. Next we show that removing the variance σ^2 input from all networks (i.e., a blind formulation) result in a very significant 1.24dB PSNR drop. Finally, in the last row, we show that the drop is even higher (1.28dB) when both fusion and refinement are disabled.

Capacity Distribution. In Table 1b we report how EMVD is affected by the number of filters and convolutions in each stage. Our experiments indicate that it is beneficial to dedicate more capacity to the denoising CNN (as denoising is the most difficult stage), while the capacity of both fusion and refinement CNNs can be reduced without significantly impacting performance.

GFLOPs	$\bar{p} \times \bar{p}$	$p \times p$	PSNR / SSIM
2542.86	1×1	1×1	44.51 / 0.9897
2544.25	3×3	1×1	**44.58 / 0.9899**
2545.49	3×3	3×3	**44.58 / 0.9899**
2546.73	5×5	1×1	44.48 / 0.9897
2550.44	5×5	5×5	44.51 / 0.9897

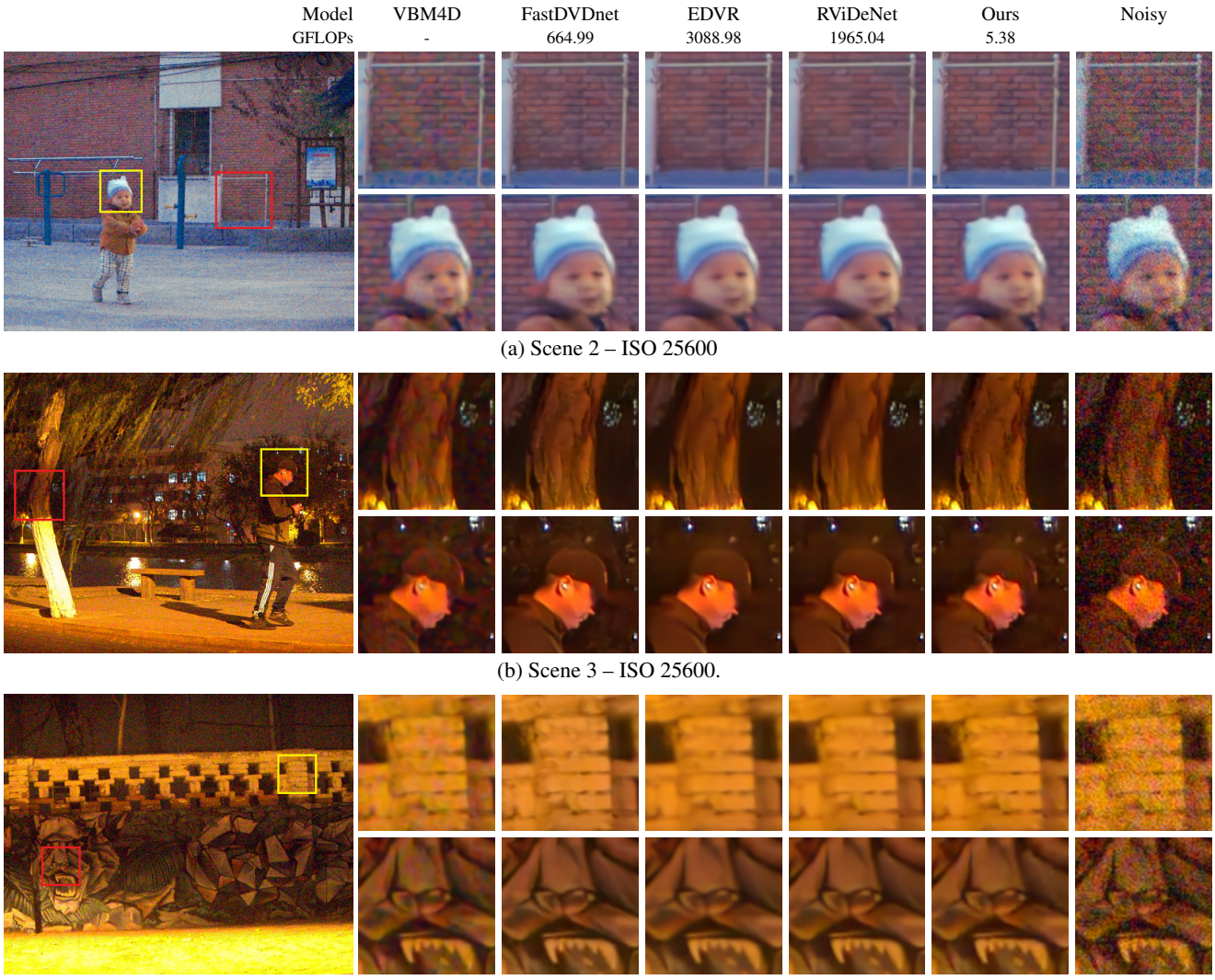
Table 2: Ablation on the fusion kernel sizes of the proposed EMVD evaluated on the raw CRVD dataset [42].

Fusion Recurrence. We use the previous fused image $\bar{y}_{t-1}$ as input of the fusion is preferable than using the previous output frame $\hat{y}_{t-1}$ or noisy frame z_{t-1} as we aim to maximally reducing the noise without removing image details. In fact, using the noisy z_{t-1} will at most decrease the noise variance by a factor of 2, effectively reducing our recurrent model to a multi-frame one. Then, as reported in Table 1a, using the output $\bar{y}_{t-1}$ is also suboptimal. This might be counter-intuitive, but since denoising is applied to $\bar{y}_{t-1}$, then the recurrent variance (9) would no longer admit a closed-form definition (because denoising is nonlinear), and also some of the high-frequency information in the image might be oversmoothed or even missing.

Fusion Prediction. In Table 2, we compare validate kernel-predicting fusion (7) using various kernel sizes. The first row correspond to the top-performing baseline model defined in Table 1b with element-wise fusion (i.e., 1×1 kernels). We compare different sizes (up to 5×5) for the kernels $\bar{p} \times \bar{p}$ and $p \times p$ applied to the previous fused image and to the current noisy image, respectively. We observe that it is more beneficial to use large kernels on the previous image, which is in fact where motion compensation is needed, and that this strategy improves PSNR by ~0.1dB PSNR at a relatively limited increase in complexity

4.3. Results

Table 3 and Fig. 6 show objective results for all compared models. We denote with † low-complexity implemen-



(a) Scene 2 – ISO 25600

(b) Scene 3 – ISO 25600.

(c) Scene 9 – ISO 25600.

Figure 5: Visual comparisons on SONY IMX385 1080p videos from the CRVD dataset [42]. The proposed EMVD exhibits more details and better noise suppression than more complex state-of-the-art methods in both static and dynamic regions.

Model	GFLOPs	raw	sRGB
EDVR	3088.98	**44.71 / 0.9902**	**40.89 / 0.9838**
RViDeNet	1965.04	44.08 / 0.9881	40.03 / 0.9802
FastDVDnet	664.99	44.30 / 0.9891	39.91 / 0.9812
Ours	79.52	44.05 / 0.9890	39.53 / 0.9796
FastDVDnet†	22.16	42.25 / 0.9806	37.43 / 0.9693
Ours	5.38	42.63 / 0.9851	38.27 / 0.9722
VBM4D	-	-	35.20 / 0.9577

Table 3: Objective performance on the CRVD dataset [42]. Reduced complexity is denoted by †.

tations which we obtain by evenly reducing the number of convolutional layers and channels. Our experiments indicate that the proposed EMVD significantly outperforms all compared methods when GFLOPs is lower than 100, and the improvement in PSNR increases to more than 1dB as the complexity decreases. Note that we do not provide results for RViDeNet and EDVR with complexity lower than 100 GFLOPs because such models fail to converge in that range. Details of the network structures of the proposed method at varying level of complexity can be found in Table 1b. More technical implementation details are discussed in the supplementary materials.

Interestingly, EMVD is even able to maintain a good margin over state-of-the-art methods with significantly higher complexity. For instance, if we compare the ~79 GFLOPs EMVD against a 25× larger RViDeNet we observe only a 0.03dB loss in PSNR. This confirms the

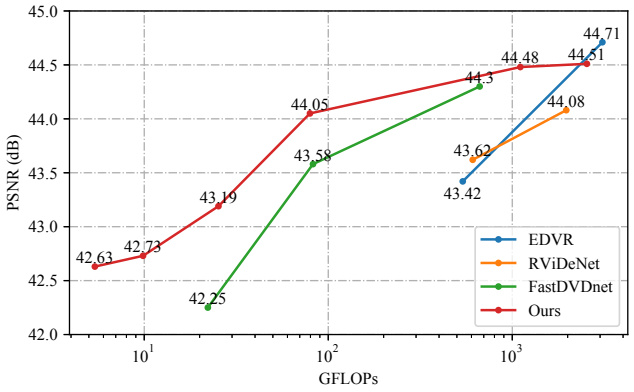


Figure 6: Performance (PSNR) of different models at various complexity levels (GFLOPs) on the raw CRVD dataset [42].

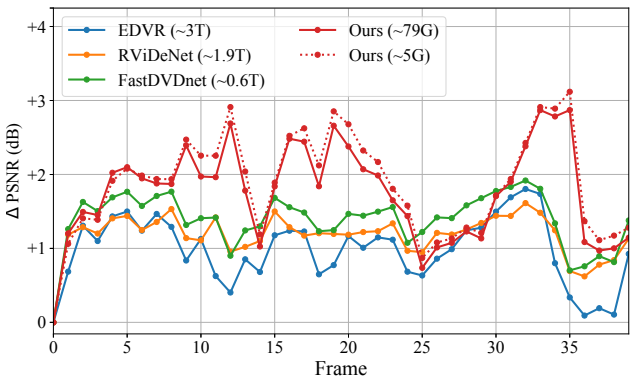


Figure 7: Frame-by-frame increase of PSNR with respect to the first frame of the sequence using the SRVD dataset [10, 42]. Complexity (FLOPs) is reported in parenthesis.

ability of our method to operate under stringent computational budgets without significant loss in performance and image quality. In Fig. 5 we show the visual comparison for several test videos in the CRVD dataset captured at ISO 25600. The bricks and trees in the background indicate that EDVR and FastDVDnet generate better details than RViDeNet. The proposed EMVD generates the most pleasing visual results in both static and dynamic regions despite having a complexity 573× lower than EDVR and 364× lower than RViDeNet.

In Fig. 7 we analyze the temporal behavior of the compared models using the SRVD dataset (MOT17-01 synthesized ISO 25600) [10, 42]. In particular, the plot shows the frame-by-frame PSNR difference (Δ) with respect to the initial frame of the sequence. We note that the Δ PSNR of RViDeNet, FastDVDnet, and EDVR is –on average– stable after an initial warm-up period. As a matter of fact, performance of multi-frame methods is bounded by the amount of frames that these models are designed to process at any given time. Differently, the proposed EMVD is a recurrent method, thus by accumulating long-term temporal dependencies it is able to achieve a significantly higher Δ PSNR in the majority of the sequence. However, multi-frame methods have a slight advantage in dynamic scenes (i.e., frame 25–30) because they can access future frames. Nevertheless, even in these cases the proposed EMVD is able to recover very quickly, as it outperforms all other methods by more than 2dB PSNR after processing only a few frames.

Finally, Table 4 reports on-chip running time and memory usage profiled on a commercial SoC (Huawei P40 Pro Smartphone) using the AI benchmark tool [21]. Results demonstrate that our method can attain real-time performance (~30fps) while still outperforming the reference low-complexity method FastDVDNet in terms of computational requirements (4.9× faster inference, 6.5× less memory) as well as objective performance (0.84dB better PSNR).

Model	GFLOPs	Time (ms)	DDR (MB)	PSNR / SSIM
FastDVDnet†	22.16	177	724	37.43 / 0.9693
Ours	5.38	36	112	**38.27 / 0.9722**

Table 4: Running time and DDR memory required to process a single-precision 720p sequence on a Huawei P40 Pro. Models have been profiled with the AI benchmark tool [21].

5. Conclusions

In this work we have proposed EMVD, an efficient video denoising method which recursively exploit the spatio-temporal correlation inherently present in natural videos through multiple cascading processing stages applied in a recurrent fashion, namely temporal fusion, spatial denoising, and spatio-temporal refinement.

This multi-stage design, coupled with learnable and invertible decorrelating transforms, allows to significantly reduce the model complexity without seriously impacting its performance. It is interesting to note that the CNNs employed in each individual stage converge to the desired behavior without explicit supervision (i.e., extra terms in the loss), hence making the proposed model straightforward to train. Further, we can gain insights on the inner workings of the model by inspecting the output at every stage (Fig. 3 and Fig. 4) which in turn allows to interpret and disentangle the effects of spatial and temporal processing.

The proposed EMVD 1) significantly outperforms other state-of-the-art video restoration methods when complexity is constrained, and 2) even remains competitive when complexity of compared models is several orders of magnitude higher (Fig. 6). Further, we have verified that EMVD achieves real-time performance (~30fps at 720p) on a commercial SoC (Table 4) with a limited memory footprint, thus demonstrating that the proposed method can be practically employed for video processing on mobile devices.

References

[1] Pablo Arias and Jean-Michel Morel. Video denoising via empirical bayesian estimation of space-time patches. *Journal of Mathematical Imaging and Vision*, 60(1):1573–7683, 2018. 1

[2] Lucio Azzari and Alessandro Foi. Gaussian-Cauchy mixture modeling for robust signal-dependent noise estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5357–5361, 2014. 2

[3] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11036–11045, 2019. 2

[4] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65, 2005. 2

[5] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005. 1, 2

[6] Antoni Buades and Joan Duran. CFA video denoising and demosaicking chain via spatio-temporal patch-based filtering. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):4143–4157, 2020. 3, 6

[7] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4778–4787, 2017. 2

[8] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. *European Conference on Computer Vision (ECCV)*, pages 171–187, 2020. 2

[9] Chen Chen, Qifeng Chen, Minh N Do, and Vladlen Koltun. Seeing motion in the dark. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3185–3194, 2019. 2

[10] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3291–3300, 2018. 5, 8

[11] Michele Claus and Jan van Gemert. ViDeNN: Deep blind video denoising. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 2

[12] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 1, 2

[13] Alex Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. A non-local cnn for video denoising. In *IEEE International Conference on Image Processing (ICIP)*, pages 2409–2413, 2019. 2

[14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 184–199. Springer, 2014. 2

[15] Jana Ehmann, Lun-Cheng Chu, Sung-Fang Tsai, and Chia-Kai Liang. Real-time video denoising on mobile phones. In *IEEE International Conference on Image Processing (ICIP)*, pages 505–509, 2018. 1, 2

[16] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. 1, 2

[17] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3476–3485, 2019. 2

[18] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics*, 35(6), 2016. 2

[19] Clement Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[20] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1712–1722, 2019. 2

[21] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. AI benchmark: All about deep learning on smartphones in 2019. In *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2019. https://ai-benchmark.com. 8

[22] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3224–3232, 2018. 2

[23] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3482–3492, 2020. 2

[24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*, 2014. 5

[25] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S. Huang. Non-local recurrent network for image restoration. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1680–1689, 2018. 1, 2

[26] Jiaming Liu, Chi-Hao Wu, Yuzhi Wang, Qin Xu, Yuqian Zhou, Haibin Huang, Chuan Wang, Shaofan Cai, Yifan Ding, Haoqiang Fan, and Jue Wang. Learning raw image denoising with bayer pattern unification and bayer preserving augmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2070–2077, 2019. 5

[27] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restora-

tion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 886–88609, 2018. 1, 2

[28] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Transactions on image processing*, 21(9):3952–3966, 2012. 1, 2, 5

[29] Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. *IEEE Transactions on Image Processing*, 22(1):91–103, 2013. 2

[30] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2502–2510, 2018. 1, 2, 4, 5

[31] MindSpore. https://www.mindspore.cn, 2020. 5

[32] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6626–6634, 2018. 2

[33] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 6

[34] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[35] Matias Tassano, Julie Delon, and Thomas Veit. DVDnet: A fast network for deep video denoising. In *IEEE International Conference on Image Processing (ICIP)*, pages 1805–1809, 2019. 2

[36] Matias Tassano, Julie Delon, and Thomas Veit. FastDVDnet: Towards real-time deep video denoising without flow estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5

[37] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. TDAN: Temporally-deformable alignment network for video super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3369, 2020. 2

[38] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 0–0, 2019. 2, 5

[39] Yuzhi Wang, Haibin Huang, Qin Xu, Jiaming Liu, Yiqun Liu, and Jue Wang. Practical deep raw image denoising on mobile devices. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[40] Moritz Wolter, Shaohui Lin, and Angela Yao. Neural network compression via learnable wavelet transforms. In *Artificial Neural Networks and Machine Learning (ICANN)*, pages 39–51, 2020. 3

[41] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3106–3115, 2019. 2

[42] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2301–2310, 2020. 1, 2, 5, 6, 7, 8

[43] 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, 2017. 1, 2