ReMoNet: Recurrent Multi-Output Network for Efficient Video Denoising

Liuyu Xiang[1,2], Jundong Zhou[1,2], Jirui Liu[1], Zerun Wang[1,2],
Haidong Huang[3], Jie Hu[3], Jungong Han[4], Yuchen Guo[1]*, Guiguang Ding[1,2]*
[1] Beijing National Research Center for Information Science and Technology (BNRist)
[2] School of Software, Tsinghua University, Beijing, China
[3] OPPO Inc, Guangdong, China.
[4] Computer Science Department, Aberystwyth University, SY23 3FL, UK
{xiangly17,zhoujd21,wang-zr19}@mails.tsinghua.edu.cn, liujirui2000@outlook.com,
{huanghaidong,hujie1}@oppo.com, jungonghan77@gmail.com, yuchen.w.guo@gmail.com, dinggg@tsinghua.edu.cn

Abstract

While deep neural network-based video denoising methods have achieved promising results, it is still hard to deploy them on mobile devices due to their high computational cost and memory demands. This paper aims to develop a lightweight deep video denoising method that is friendly to resource-constrained mobile devices. Inspired by the facts that 1) consecutive video frames usually contain redundant temporal coherency, and 2) neural networks are usually over-parameterized, we propose a multi-input multi-output (MIMO) paradigm to process consecutive video frames *within one-forward-pass*. The basic idea is concretized to a novel architecture termed Recurrent Multi-output Network (ReMoNet), which consists of recurrent temporal fusion and temporal aggregation blocks and is further reinforced by similarity-based mutual distillation. We conduct extensive experiments on NVIDIA GPU and Qualcomm Snapdragon 888 mobile platform with Gaussian noise and simulated Image-Signal-Processor (ISP) noise. The experimental results show that ReMoNet is both effective and efficient on video denoising. Moreover, we show that ReMoNet is more robust under higher noise level scenarios.

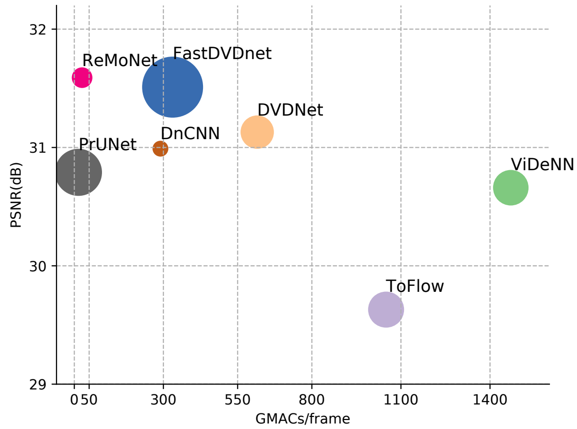


Figure 1: Comparison of PSNR and computational cost on Set8 with noise level $\sigma = 30$. The radius represents #parameters. Compared to existing video denoising methods, the proposed ReMoNet achieves state-of-the-art performance with much lower MACs.

Introduction

In recent years, a plethora of deep learning-based approaches have achieved remarkable progress on video denoising (Davy et al. 2018; Xue et al. 2019; Claus and van Gemert 2019; Tassano, Delon, and Veit 2020). Most of them focus on how to fully exploit the temporal coherency and inter-frame relationships, so that better denoising performance can be achieved. Existing video denoising methods can be broadly categorized into two types according to their temporal modeling strategies: explicit and implicit temporal modeling. The former can usually be formulated as $y_i = f(w_i(x_{i-T}), ..., x_i, ..., w_i(x_{i+T}))$, where $w_i(x_j)$ denotes warping from frame j to frame i using explicit motion estimation such as optical flow, and f denotes the denoising function parameterized by deep neural networks (Xue et al. 2019; Tassano, Delon, and Veit 2019). The latter can be formulated as $y_i = f(x_{i-T}, ..., x_{i+T})$, where no explicit flow estimation is used and temporal information is implicitly extracted and fused by the deep network f (Zhang et al. 2018; Tassano, Delon, and Veit 2020). Since there is no extra computation for flow estimation, implicit temporal modeling methods usually have less computational cost compared to explicit ones. While various temporal modeling methods have been proposed, few efforts have been made to develop lightweight models for demanding mobile video applications. Even the most lightweight existing SOTA method FastDVDNet (Tassano, Delon, and Veit 2020) still has 331G MACs (Multi-Adds) per 540P (960*540) frame, and is far from practical video applications on edge.

In this paper, we take a step forward and reduce the MACs per frame from 331G to 26G with comparable or even superior performance (See Figure. 1). Our method is inspired by two key observations. Firstly, consecutive video frames usually share similar content and contain visual redundancy (Xiao et al. 2015; Buckler et al. 2018). Secondly, neural networks are usually over-parameterized (Arora, Cohen, and Hazan 2018; Frankle and Carbin 2018; Chen et al. 2019; Allen-Zhu, Li, and Song 2019), as evidenced by the findings that neural networks can maintain their performance even

*Corresponding Authors

Method	Temp.	Blind	High perf.	High efficiency	ISP exp.	Mobile exp.
DnCNN (Zhang et al. 2017)	✗	✓	✓	✗	✗	✗
PrUNet (Wang et al. 2020)	✗	✓	✗	✓	✓	✓
ToFlow (Xue et al. 2019)	✓	✓	✓	✗	✗	✗
ViDeNN (Claus and van Gemert 2019)	✓	✓	✓	✗	✗	✗
DVDNet (Tassano, Delon, and Veit 2019)	✓	✗	✓	✗	✗	✗
FastDVD (Tassano, Delon, and Veit 2020)	✓	✗	✓	✗	✗	✗
ReMoNet (Ours)	✓	✓	✓	✓	✓	✓

Table 1: Comparison of ReMoNet and existing denoising methods.

when over 70% parameters are pruned (Liu et al. 2017; Zhu and Gupta 2017; Frankle and Carbin 2018).

Based on these observations, we propose a simple yet effective Multi-input Multi-output (MIMO) formulation:

$$[y_{i-T}, ..., y_{i+T}] = f(x_{i-T}, ..., x_{i+T})$$

We assume that the neural network has enough capacity to process video sequences with shared visual redundancy **within one forward pass**. We further show that with proper design, the MIMO-style denoising network can achieve competitive performance as well as high computational efficiency.

To be more specific, we propose Recurrent Multi-output Network (ReMoNet). It has two major components: the Recurrent Temporal Fusion (RTF) block and Multi-output Aggregation (MOA) block. The RTF block has the structure of a lightweight U-Net. It recurrently extracts and fuses the temporal information from the video sequences with a MIMO strategy. Then the MOA block also works in a MIMO manner where the hidden features of multiple frames are processed simultaneously within a single forward pass. Finally, we reinforce the multi-output process by similarity-based mutual distillation that further improves the denoising performance. Since the whole ReMoNet takes multiple frames as inputs and produces multiple frames in parallel, it significantly reduces the computational cost per frame and accelerates the denoising speed.

We carry out extensive experiments on two video denoising benchmark datasets: Set8 and DAVIS, with two types of noises: Gaussian noise and simulated ISP noise, on two popular platforms: NVIDIA GPU and Qualcomm Snapdragon mobile platform, to validate the effectiveness of the proposed method. We show that ReMoNet is able to achieve competitive performance under Gaussian noise and outperforms all baselines under realistic noise. More importantly, it has only **7.9% MACs and 32.4% parameters**, compared to the previous most lightweight SOTA FastDVDNet (Tassano, Delon, and Veit 2020). The experimental results also show that the ReMoNet performs even better when the noise level becomes higher. A brief comparison of ReMoNet and existing methods is shown in Table. 1. We compare on the following indicators: 1) whether has temporal modeling for consecutive frames, 2) whether is blind denoising (i.e. unknown of noise level), 3) whether has high performance (comparable to SOTA methods like FastDVDNet), 4) whether has high efficiency, 5) whether conducts experiments on ISP noise, 6) whether conducts experiments on mobile platforms. Table. 1 shows the significant superiority of ReMoNet over existing methods.

Related Work

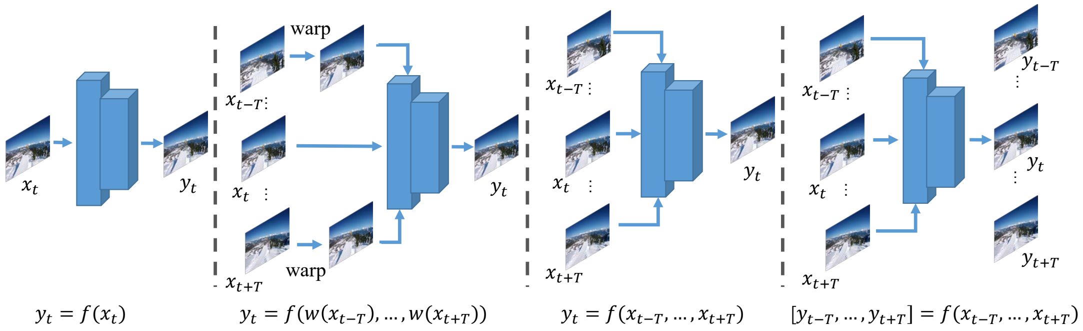
Image Denoising Methods

Image denoising has long been investigated in the literature (Motwani et al. 2004; Fan et al. 2019). Traditional methods can be roughly categorized into two types: spatial domain and transform domain. The former (Buades, Coll, and Morel 2005; Beck and Teboulle 2009) exploits the correlations between similar pixels or patches, among which Non-local Means (NLM) (Buades, Coll, and Morel 2005) is the representative. The latter (Mihcak et al. 1999; Starck, Candès, and Donoho 2002; Dabov et al. 2007) usually denoise in the Fourier or wavelet domain among which BM3D (Dabov et al. 2007) is the representative.

With the development of deep learning, CNN-based methods gradually become prevalent due to their outstanding performance. DnCNN (Zhang et al. 2017) learns the residual mapping from input to noise signal. FFDNet (Zhang, Zuo, and Zhang 2018) further introduces a controllable noise map to cope with various noise levels. There are also attempts for real-world noise removal. Various realistic noisy datasets with ground-truth, such as DND (Plotz and Roth 2017) and SIDD (Abdelhamed, Lin, and Brown 2018), are collected to facilitate practical applications. A practical raw image denoising network is also introduced (Wang et al. 2020), which can be deployed on mobile devices.

Video Denoising Methods

Traditional video denoising methods usually generalize image denoising methods to their video processing counterpart. One representative is VBM4D (Maggioni et al. 2012), which generalizes BM3D to 4D dimension and results in better temporal consistency. Deep learning methods tackle this issue either with explicit or implicit temporal modeling. Among these methods, Non-Local Net (Davy et al. 2018) extends the idea of NLM (Buades, Coll, and Morel 2005) and searches similar adjacent image patches for training. ToFlow (Xue et al. 2019), DVDNet (Tassano, Delon, and Veit 2019) and JointLearn (Yu et al. 2020) all use various optical flow computation (Horn and Schunck 1981; Weinzaepfel et al. 2013) for motion estimation, so that adjacent frames are warped to provide temporal similarities. ViDeNN (Claus and van Gemert 2019) and FastDVDNet (Tas-


x_t
y_t
warp
x_{t-T}
x_t
x_{t+T}
warp
y_t
x_{t-T}
x_t
x_{t+T}
y_t
x_{t-T}
x_t
x_{t+T}
y_{t-T}
y_t
y_{t+T}
$y_t = f(x_t)$
$y_t = f(w(x_{t-T}), \ldots, w(x_{t+T}))$
$y_t = f(x_{t-T}, \ldots, x_{t+T})$
$[y_{t-T}, \ldots, y_{t+T}] = f(x_{t-T}, \ldots, x_{t+T})$

(a) Single-frame (b) Explicit temporal modeling (c) Implicit temporal modeling (d) Multi-input Multi-output

Figure 2: Comparison of different temporal modeling types. The proposed Multi-input Multi-output modeling enables both effective temporal modeling and computational efficiency.

sano, Delon, and Veit 2020), on the other hand, use two-stage spatial-temporal architecture without explicit motion estimation. There are also methods that directly denoise on RAW video inputs. RViDeNet (Yue et al. 2020) introduces a dynamic raw video denoising dataset with groundtruth and proposes a novel denoising architecture. EMVD (Maggioni et al. 2021) further proposes an efficient and effective video denoising method. In this paper, we mainly focus on video denoising in the sRGB space.

Although progress has been made in video denoising, most methods require a large computational cost. In this sense, great gaps still exist between current algorithms and practical mobile-friendly applications.

Preliminary and Motivation

Given a noisy video, we usually have three solutions for denoising: 1) apply single-frame image denoising algorithms without using any temporal information (Figure. 2 (a) $y_i = f(x_i)$). 2) Use explicit temporal modeling. (Figure. 2 (b) $y_i = f(w_i(x_{i-T}), ..., x_i, ..., w_i(x_{i+T}))$) 3) Use implicit temporal modeling (Figure. 2 (c) $y_i = f(x_{i-T}, ..., x_{i+T})$).

We notice that while the temporal coherency in videos provides visual similarities and boosts the performance, it also requires the network to process large amounts of redundant content. In other video applications such as video coding, redundancy reduction is usually taken *in the input space*, where consecutive similar frames are compressed to (or predicted by) one representative frame (Sousa 2000; Wiegand et al. 2003). However, this is not applicable in our pixel-to-pixel denoising task. In this work, we propose to reduce the redundant computation *in the output space* by the Multi-input Multi-output strategy. If we take the advantage of over-parameterization in neural networks and process consecutive frames *during one forward pass*, the computation on those redundant visual content can be largely reduced. Taken a denoising network with $2T+1$ temporal input for instance, when converting to its MIMO version, only slight changes are needed (i.e. change the number of channels in the last convolutional layer) which will bring almost negligible extra parameters (usually less than 1%). Meanwhile, since the MIMO-network denoises $2T+1$ frames during one forward pass, theoretically it could be almost $2T+1$ times faster. This leaves us one question to answer, do neural networks have enough capacity to process multiple frames concurrently, or in other words, can MIMO-network maintain competitive performance? We will elaborate in the next section that, with proper design, the proposed ReMoNet is able to achieve both efficiency and effectiveness.

Proposed Method

Overview

We consider $2T + 1$ consecutive frames as inputs following the standard video denoising paradigm and denote them as $[x_{i-T}, ..., x_{i+T}]$ where $x_i \in R^{C\times H\times W}$ and C usually equals to 3 when x is in the sRGB space. As shown in Figure. 3, the proposed ReMoNet consists of two blocks: the RTF block f_{RTF} recurrently fuses the temporal information from consecutive frames and extracts the latent feature frames $z_t \in R^{C_z\times H\times W}$. Then the MOA block f_{MOA} works in a multi-input multi-output style to further aggregate adjacent temporal information and yield the processed frames.

Recurrent Temporal Fusion Block

To fully extract the temporal relationships in adjacent video frames, various spatial-temporal implicit fusion methods have been proposed. According to the categorization in (Caballero et al. 2017), there are usually two types of temporal fusion shown in Figure. 4 (a-b). The fast temporal fusion usually concatenates all input frames together along the channel dimension, and thus temporal information collapses at the first layer. The slow temporal fusion usually merges $2K + 1$ frames in groups smaller than the input number of frames $2T + 1$, and gradually fuses along the temporal axis. This is in some way, equivalent to 3D convolution with a temporal kernel size $2K+1$. As demonstrated by FastDVDNet (Tassano, Delon, and Veit 2020) that slow fusion usually performs superior to fast fusion, we take one step further, and propose a recurrent temporal fusion. Apart from gradually fusing temporal information with sliding window size $2K+1$, we also use a simple recurrent structure to keep track of the temporal visual relationships. To be more specific, the RTF Block has a structure of a lightweight tiny UNet (Ronneberger, Fischer, and Brox 2015) with two downsampling and two upsampling stages, whose structure will be elaborated in the supplementary material. For sliding window at

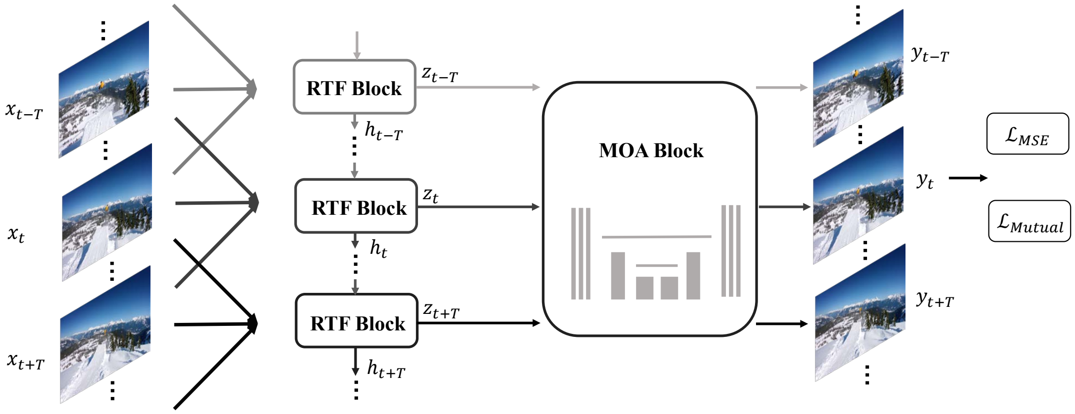
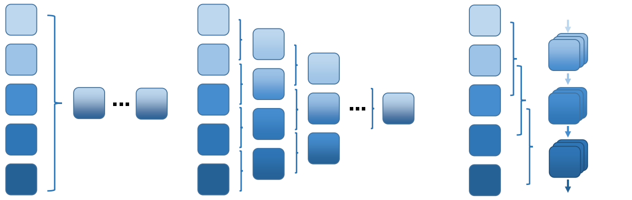


Figure 3: Illustration of ReMoNet. It mainly consists of two blocks: the Recurrent Temporal Fusion (RTF) block recurrently fuses temporal information while the Multi-output Aggregation block (MOA) aggregates them in a multi-output manner.



(a) Fast temporal fusion (b) Slow temporal fusion (c) Recurrent temporal fusion

Figure 4: Different types of temporal fusion (Caballero et al. 2017) (a-b) and the proposed Recurrent Temporal Fusion with Multi-ouput (c).

position t, we have:

$$[z_t, h_t] = f_{RTF}(\text{concat}([x_{t-K}, ..., x_{t+K}, z_{t-1}, h_{t-1}])$$

where $h_i \in R^{L \times H \times W}$ is the recurrent hidden state.

Note that z_t contains essential information for restoration and usually has the same shape of input frame $R^{C \times H \times W}$, this is also, to some extent, collapsing temporal information (like fast temporal fusion do), and can be improved. Thus we propose to extend the vanilla RTF block into its MIMO version and let $C_z = (2K + 1)C$, so that the RTF block takes $2K+1$ frames as inputs, and outputs $2K+1$ 'hidden frames' $z_t \in R^{(2K+1)C \times H \times W}$ at each timestep t. We show by experiments that this MIMO-style recurrent temporal fusion will be beneficial for the restoration. Furthermore, we visualize these hidden frames z_t in the supplementary material and observe that RTF-MIMO implicitly learns more diverse information compared to the non-MIMO RTF block.

Multi-output Aggregation Block

Once we have the hidden frames z_t, we notice that each z_t is produced with temporal kernel size $2K+1$ which is less than the whole sequence length $2T + 1$, and we wish to enlarge the 'temporal receptive field' to further refine the temporal knowledge in z_t. Here we again follow the MIMO paradigm for such temporal aggregation.

To be specific, we use another lightweight UNet-like network and convert it to its MIMO version.

$$[y_{t-T}, ..., y_{t+T}] = f_{MOA}(z_{t-T}, ..., z_{t+T})$$

where $y_i \in R^{C \times H \times W}$. It takes multiple $z_i \in R^{(2K+1) \times H \times W}, i \in [t - T, t + T]$ as inputs, and outputs $[y_{t-T}, ..., y_{t+T}]$. The MIMO conversion only requires changing the output channel size and will bring almost neglectable extra parameters. The benefits of such multi-output aggregation are two-folded: first, it largely accelerates the processing speed since multiple frames are processed during one forward pass. Meanwhile, it also improves the denoising performance since the locally fused temporal information in each z_t is aggregated and refined.

Similarity-based Mutual Distillation

To further improve the performance, we propose to use similarity-based mutual distillation to reinforce the temporal aggregation in the MOA block. Inspired by the success of deep mutual learning (Zhang et al. 2018), where several peer networks are trained to teach each other, we propose to train two identical ReMoNets and supervise the outputs of MOA block mutually via Similarity-Preserving distillation (Tung and Mori 2019). Since both RTF and MOA block work in a MIMO manner, we wish to make full use of the network capacity and let them absorb more information. Then the similarity-based mutual distillation provides an extra supervision signal to force the network to learn not only from ground-truth, but also from their peers. Concretely, if we have two randomly initialized ReMoNets, and $y_1, y_2 \in R^{B \times C \times H \times W}$ to be their corresponding outputs and B be the batch size, the similarity-based mutual distillation forces y_1 and y_2 to have a close similarity map within the mini-batch:

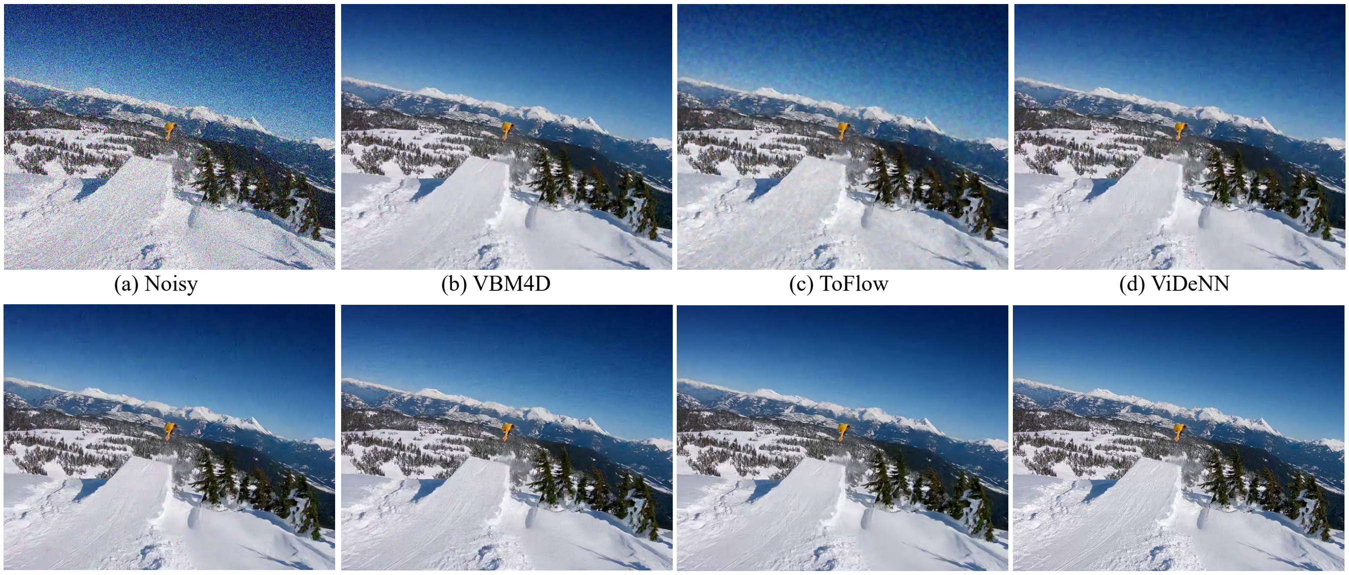
$$L_{Mutual}(y_1, y_2) = \frac{1}{B^2} ||g_1 - g_2||_F^2$$

$$g_i = norm(a_i \cdot a_i^T) \quad a_i = \text{Reshape}(y_i) \in R^{B \times CHW}$$

where $|| \cdot ||_F$ is the Frobenius norm, $norm$ denotes row-wise L2 normalization and g_i represents the visual similarity map between frames within each mini-batch. Intuitively, since the training trajectory of neural networks is stochastic, each student network may learn different information and be

Method	Comp. Cost		$\sigma = 10$		$\sigma = 20$		$\sigma = 30$		$\sigma = 40$		$\sigma = 50$	
	MACs	#Param	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
VBM4D	-	-	35.59	0.9295	32.02	0.8798	29.90	0.8340	28.50	0.8018	27.33	0.7672
DnCNN	290G	559K	36.27	0.9485	32.91	0.9045	30.99	0.8662	29.66	0.8325	28.65	0.8027
PrUNet	**14G**	1.89M	33.24	0.8763	32.40	0.8919	30.79	0.8644	23.84	0.5290	21.03	0.3672
ToFlow	1.05T	1.44M	34.34	0.9241	31.44	0.8675	29.63	0.8153	28.33	0.7664	27.26	0.7183
ViDeNN	1.47T	1.42M	34.91	0.9387	32.34	0.8981	30.66	0.8589	29.46	0.8249	28.52	0.7938
EMVD	26G	**360K**	35.41	0.9425	32.50	0.8999	30.71	0.8614	29.46	0.8273	28.53	0.7976
DVD	616G	1.33M	35.91	0.9470	32.94	0.9094	31.12	0.8740	29.85	0.8412	28.87	0.8111
F.DVD	331G	2.48M	**36.48**	**0.9531**	**33.34**	0.9169	31.51	0.8840	30.25	0.8541	29.28	0.8268
F.D-B	330G	2.48M	36.44	0.9530	33.32	0.9166	31.50	0.8836	30.23	0.8535	29.27	0.8260
ReMoNet	26G	804K	36.29	0.9528	**33.34**	**0.9179**	**31.59**	**0.8859**	**30.37**	**0.8570**	**29.44**	**0.8308**

Table 2: Results on Set8 with Gaussian noise



(a) Noisy (b) VBM4D (c) ToFlow (d) ViDeNN

(e) DVDNet (f) FastDVDNet (g) ReMoNet (h) Clean

Figure 5: Results on *snowboard* in GOPRO with Gaussian noise, noise level $\sigma = 40$. The results show that ReMoNet yields cleaner and more natural results than baselines. Zoom in for a better view.

complementary to one another. Finally, the total loss function is:

$$\mathcal{L} = \mathcal{L}_{MSE}(y_1, gt) + \mathcal{L}_{MSE}(y_2, gt) + \mathcal{L}_{Mutual}(y_1, y_2)$$

where $\mathcal{L}_{MSE}$ denotes the Mean-Square-Error between the ReMoNet's output y_i and the ground-truth frames gt. During testing time, we randomly choose one of the ReMoNets for inference, which will bring no extra computational cost.

Experiments

Datasets and metrics We use two benchmark datasets for evaluation: Set8 and DAVIS-test. Set8 consists of 4 sequences captured by GOPRO camera and 4 sequences from the Derf's Test Media collection, with a resolution of 960×540. The DAVIS-test contains 30 sequences of resolution 854×480. We use DAVIS-train for training. We test with two types of noise: Additive White Gaussian Noise (AWGN) with controlled σ and simulated ISP noise. PSNR and SSIM are used as performance metrics where Multiply-Add operations (MACs) per frame and number of parameters (#Param) are used as efficiency metrics.

Baseline methods We compare our method with state-of-the-art video denoising methods including: VBM4D (Maggioni et al. 2012) which is a representative traditional method, DnCNN (Zhang et al. 2018) and Practical UNet (Wang et al. 2020) (PrUNet) which are image denoising methods, ToFlow (Xue et al. 2019), ViDeNN (Claus and van Gemert 2019), EMVD (Maggioni et al. 2021), DVDNet (DVD) (Tassano, Delon, and Veit 2019), FastDVDNet (F.DVD) (Tassano, Delon, and Veit 2020) are all SOTA video denoising methods. Since FastDVDNet requires noise level map as inputs, we also compare with its blind version FastDVDNet-B (F.D-B). All baselines are reproduced using public available codes from authors and hyperparameters are tuned on target datasets.

Implementation Details In practice, we choose the input number of frames $2T + 1 = 5$ and temporal fusion size $2K + 1 = 3$, the RTF hidden dimension $L = 32$. More details can be found in the supplementary material.

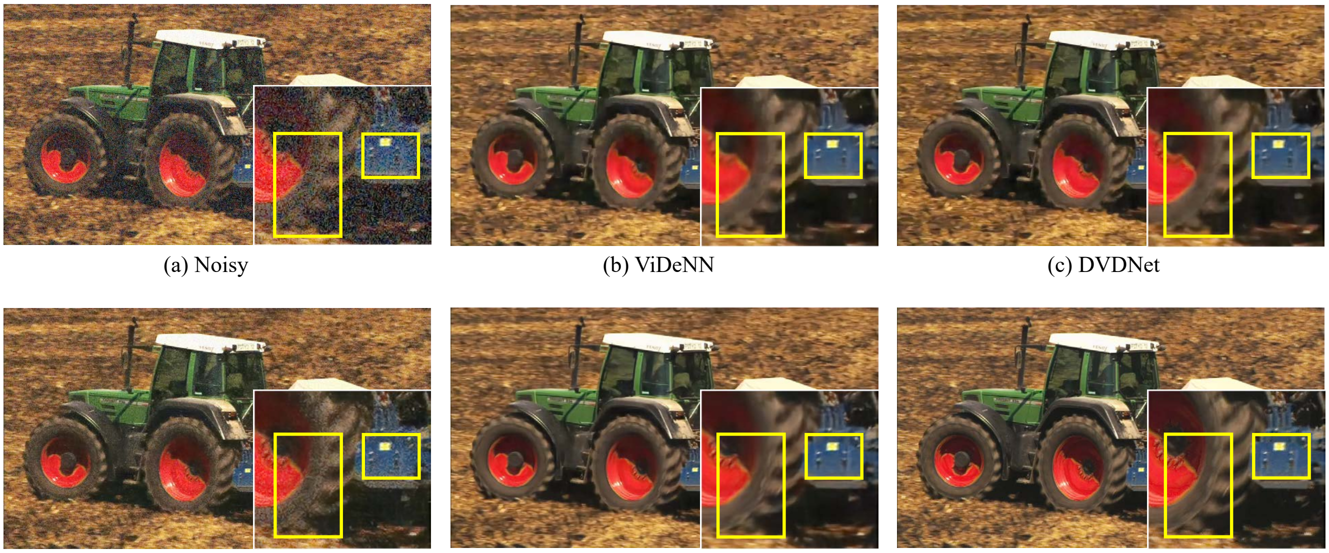
Quantitative results with Gaussian noise We compare ReMoNet with SOTAs in terms of both denoising perfor-

Method	$\sigma = 10$		$\sigma = 20$		$\sigma = 30$		$\sigma = 40$		$\sigma = 50$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
VBM4D	37.49	0.9420	34.00	0.9040	31.86	0.8690	30.18	0.8395	28.90	0.8093
DnCNN	38.63	0.9639	35.10	0.9277	33.09	0.8945	31.69	0.8648	30.62	0.8380
PrUNet	34.27	0.8827	34.46	0.9159	32.90	0.8940	24.22	0.5245	21.12	0.3466
ToFlow	36.75	0.9452	33.38	0.8934	31.30	0.8425	29.77	0.7929	28.49	0.7422
ViDeNN	37.57	0.9579	34.77	0.9235	32.99	0.8907	31.70	0.8615	30.63	0.8333
DVDNet	38.58	0.9629	35.37	0.9308	33.49	0.9017	32.16	0.8753	31.14	0.8511
FastDVDNet	38.99	0.9664	**35.78**	0.9372	33.90	0.9099	32.58	0.8851	31.58	0.8623
FastDVDNet-B	**39.04**	0.9670	35.77	0.9371	33.90	0.9096	32.59	0.8845	31.58	0.8613
ReMoNet	38.97	**0.9672**	35.77	**0.9380**	**33.93**	**0.9114**	**32.64**	**0.8872**	**31.65**	**0.8651**

Table 3: Results on DAVIS-test with Gaussian noise.

Method	$\sigma = 0.01$		$\sigma = 0.02$		$\sigma = 0.03$		$\sigma = 0.04$		$\sigma = 0.05$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DnCNN	32.48	0.8994	30.39	0.8489	29.05	0.8087	28.07	0.7751	27.30	0.7470
ViDeNN	31.88	0.8958	30.24	0.8551	29.10	0.8199	28.24	0.7899	27.56	0.7641
DVDNet	31.45	0.8943	29.93	0.8544	28.85	0.8220	28.02	0.7941	27.35	0.7694
FastDVDNet	32.45	0.8997	30.57	0.8653	28.77	0.8096	27.00	0.7298	25.28	0.6437
ReMoNet	**32.57**	**0.9083**	**30.67**	**0.8692**	**29.48**	**0.8376**	**28.62**	**0.8108**	**27.96**	**0.7875**

Table 4: Results on Set8 with simulated ISP noise. (Pixel value ranging from 0 to 1.)



(a) Noisy

(b) ViDeNN

(c) DVDNet

(d) FastDVDNet

(e) ReMoNet

(f) Clean

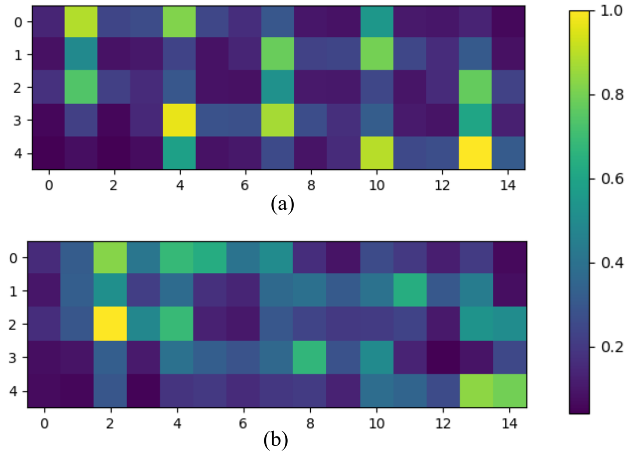
Figure 6: Results on *tractor* in Derf with simulated ISP noise, noise level $\sigma = 0.05$. We observe that ReMoNet recovers clearer details than baseline methods under realistic noise distribution (See the yellow bounding boxes). Zoom in for a better view.

mance and computational cost (calculated on 960*540 resolution) on two benchmark datasets: Set8 and DAVIS-test and the results are shown in Table 2 and 3. It can be observed that, in terms of computational efficiency, ReMoNet is significantly superior to most video denoising methods. Its MACs (26G) is two orders of magnitude lower than some video denoising methods such as ToFlow (1.05T) and ViDeNN (1.47T). Moreover, it is only 7.9% compared to previous SOTA FastDVDNet (331G). Meanwhile, when compared with single-frame methods PrUnet and DnCNN, ReMoNet has comparable MACs or #Param with much higher PSNR and SSIM. Meanwhile, in terms of denoising performance, the results show that ReMoNet outperforms all baseline methods under all noise levels except FastDVDNet. The ReMoNet performs comparably with FastDVDNet when the noise level is low ($\sigma = 10, 20$) and is superior to all baselines when the noise level is high ($\sigma = 30, 40, 50$). We also observe that the higher the noise level, the larger improvements ReMoNet will get, demonstrating ReMoNet's robustness under severe noise corruption.

Quantitative results with simulated ISP noise To further verify the generalizability of the proposed method, we compare ReMoNet with four baselines under the simulated ISP noise. The results in Table. 4 show that ReMoNet outperforms all baselines in this scenarios. More importantly, we again observe that the gap between ReMoNet and the sec-

Method	$\sigma = 10$		$\sigma = 20$		$\sigma = 30$		$\sigma = 40$		$\sigma = 50$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
w/o Recur	36.13	0.9503	33.10	0.9130	31.30	0.8792	30.04	0.8485	29.09	0.8212
w/o MO	36.18	0.9515	33.20	0.9156	31.45	0.8835	30.23	0.8545	29.30	0.8282
w/o MOA	36.10	0.9505	33.08	0.9134	31.29	0.8798	30.04	0.8493	29.09	0.8220
w/o MuD	36.20	0.9521	33.25	0.9164	31.49	0.8836	30.26	0.8538	29.33	0.8268
All	36.29	0.9528	33.34	0.9179	31.59	0.8859	30.37	0.8570	29.44	0.8308

Table 5: Ablation Study on Set8. The results show that all components make indispensable contributions.



(a)

(b)

Figure 7: Perturbation analysis on MOA with (b) and without (a) mutual distillation. Horizontal and vertical axis denote index of z_j and y_i respectively. It shows that mutual distillation enlarges the temporal range of each z_j's contribution to the output y_i.

ond best performance becomes larger as the noise level increases. The results not only verifies that ReMoNet generalizes well to sophisticated real-world noise distribution, but is also much more robust under high-level noise corruption.

Quantitative results on mobile platform In order to verify the efficiency **on mobile devices**, we compare the ReMoNet with the previous most lightweight SOTA FastDVDnet (Tassano, Delon, and Veit 2020). The experiments are conducted on the **Qualcomm Snapdragon 888 mobile platform**, where its GPU is used for inference. The input resolution is 360 × 640. The result shows that FastDVDNet requires 927ms to process one frame. Meanwhile, ReMoNet only requires 446ms to process five frames, that is, 89.2ms per frame, which is **11 times faster** than FastDVDNet. The inference speed on the mobile platform further verifies the efficiency of ReMoNet in practical applications.

Qualitative results We first compare the qualitative results of Gaussian denoising on GOPRO dataset shown in Figure. 5. From the results, we observe that most previous methods still result in incoherent sky, some of which are filled with severe noise (ToFlow, ViDeNN). In contrast, ReMoNet is able to restore a clear and coherent sky which is much more natural. We also provide the visual comparison under the simulated ISP noise shown in Figure. 6. From the *tractor* results, we observe that only ReMoNet clearly recovers the two black components in the blue mechanical device (the yellow bounding box on the right). Also note that all baseline methods fail to recover the complicated texture of the tyre, where ReMoNet restores the texture closer to the original ground-truth (the yellow bounding box on the left).

Ablation study We further conduct an ablation study to verify the effectiveness of each component. We compare ReMoNet with the following variants: 1) w/o Recur, where recurrent temporal fusion is replaced by slow fusion. 2) w/o MO, where MIMO recurrent temporal fusion is replaced by non-MIMO version. 3) w/o MOA, where MOA block is discarded and non-MIMO recurrent temporal fusion's output is used as final output. 4) w/o MuD, where mutual distillation is discarded. The results in Table 5 illustrate that all components are indispensable for the overall performance. Moreover, the results show that RTF and MOA blocks contribute around 0.16dB for $\sigma = 10$ and 0.35dB for $\sigma = 50$, both of which are critical to the high noise-level robustness.

Further analysis on MOA We further investigate how MIMO and mutual distillation work in the MOA block via perturbation analysis. The MOA block takes 5 hidden frames $[z_{-2}, ..., z_2]$ as inputs and yields 5 processed frames $[y_{-2}, ..., y_2]$. We define the perturbation of y_i with respect to z_j as: $s(y_i, z_j)$, that is, how each y_i would react when z_j changes. In practice, we replace each z_j with $z_k, k \neq j$ and see the resulting y_i^{jk}, then $s(y_i, z_j) = \sum_k MSE(y_i, y_i^{jk})$. We plot the results of $s(y_i, z_j)$ with and without mutual distillation on GOPRO shown in Figure. 7. First, the results show that the network automatically learns to absorb information from adjacent positions. Furthermore, we find that without mutual distillation, z_j mostly contributes to its neighborhood y_i while the mutual learning guides the z_j to make impacts on a broader range of y_i.

Conclusion

In this paper, we first analyze the current temporal modeling methods and propose a novel Multi-input Multi-output (MIMO) strategy which is both efficient and effective. Following the MIMO spirit, we design the ReMoNet which consists of two components: the RTF block recurrently fuses the temporal information and the MOA block further refines in a multi-output manner. Then we train the ReMoNet with similarity-based mutual distillation so that the network can capture a broader range of temporal relationships. Extensive experiments on both GPU and mobile platforms demonstrate that the ReMoNet achieves superior performance over SOTAs with significantly less computational cost. We also carry out visualizations and ablation studies to verify the effectiveness of each component.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61925107, No. 61971260, U21B2013), the Science Fund for Creative Research Groups of the National Natural Science Funds of China (No. 62021002), National Key R&D Program of China (No. 2020AAA0105500) and OPPO research funding.

References

Abdelhamed, A.; Lin, S.; and Brown, M. S. 2018. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1692–1700.

Allen-Zhu, Z.; Li, Y.; and Song, Z. 2019. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, 242–252. PMLR.

Arora, S.; Cohen, N.; and Hazan, E. 2018. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, 244–253. PMLR.

Beck, A.; and Teboulle, M. 2009. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11): 2419–2434.

Buades, A.; Coll, B.; and Morel, J.-M. 2005. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, 60–65. IEEE.

Buckler, M.; Bedoukian, P.; Jayasuriya, S.; and Sampson, A. 2018. EVA^2: Exploiting Temporal Redundancy in Live Computer Vision. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 533–546. IEEE.

Caballero, J.; Ledig, C.; Aitken, A.; Acosta, A.; Totz, J.; Wang, Z.; and Shi, W. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4778–4787.

Chen, Z.; Cao, Y.; Zou, D.; and Gu, Q. 2019. How much over-parameterization is sufficient to learn deep relu networks? *arXiv preprint arXiv:1911.12360*.

Claus, M.; and van Gemert, J. 2019. Videnn: Deep blind video denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Dabov, K.; Foi, A.; Katkovnik, V.; and Egiazarian, K. 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8): 2080–2095.

Davy, A.; Ehret, T.; Morel, J.-M.; Arias, P.; and Facciolo, G. 2018. Non-local video denoising by cnn. *arXiv preprint arXiv:1811.12758*.

Fan, L.; Zhang, F.; Fan, H.; and Zhang, C. 2019. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1): 1–12.

Frankle, J.; and Carbin, M. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Horn, B. K.; and Schunck, B. G. 1981. Determining optical flow. *Artificial intelligence*, 17(1-3): 185–203.

Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; and Zhang, C. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, 2736–2744.

Maggioni, M.; Boracchi, G.; Foi, A.; and Egiazarian, K. 2012. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Transactions on image processing*, 21(9): 3952–3966.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3466–3475.

Mihcak, M. K.; Kozintsev, I.; Ramchandran, K.; and Moulin, P. 1999. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12): 300–303.

Motwani, M. C.; Gadiya, M. C.; Motwani, R. C.; and Harris, F. C. 2004. Survey of image denoising techniques. In *Proceedings of GSPX*, volume 27, 27–30.

Plotz, T.; and Roth, S. 2017. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1586–1595.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Sousa, L. 2000. General method for eliminating redundant computations in video coding. *Electronics Letters*, 36(4): 306–307.

Starck, J.-L.; Candès, E. J.; and Donoho, D. L. 2002. The curvelet transform for image denoising. *IEEE Transactions on image processing*, 11(6): 670–684.

Tassano, M.; Delon, J.; and Veit, T. 2019. Dvdnet: A fast network for deep video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*, 1805–1809. IEEE.

Tassano, M.; Delon, J.; and Veit, T. 2020. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1354–1363.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1365–1374.

Wang, Y.; Huang, H.; Xu, Q.; Liu, J.; Liu, Y.; and Wang, J. 2020. Practical Deep Raw Image Denoising on Mobile Devices. In *European Conference on Computer Vision*, 1–16. Springer.

Weinzaepfel, P.; Revaud, J.; Harchaoui, Z.; and Schmid, C. 2013. DeepFlow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*, 1385–1392.

Wiegand, T.; Sullivan, G. J.; Bjontegaard, G.; and Luthra, A. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7): 560–576.

Xiao, J.; Liao, L.; Hu, J.; Chen, Y.; and Hu, R. 2015. Exploiting global redundancy in big surveillance video data for efficient coding. *Cluster Computing*, 18(2): 531–540.

Xue, T.; Chen, B.; Wu, J.; Wei, D.; and Freeman, W. T. 2019. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8): 1106–1125.

Yu, S.; Park, B.; Park, J.; and Jeong, J. 2020. Joint learning of blind video denoising and optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 500–501.

Yue, H.; Cao, C.; Liao, L.; Chu, R.; and Yang, J. 2020. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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 transactions on image processing*, 26(7): 3142–3155.

Zhang, K.; Zuo, W.; and Zhang, L. 2018. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9): 4608–4622.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4320–4328.

Zhu, M.; and Gupta, S. 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.