PRACTICAL REAL VIDEO DENOISING WITH REALISTIC DEGRADATION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing video denoising methods typically assume noisy videos are degraded from clean videos by adding Gaussian noise. However, deep models trained on such a degradation assumption will inevitably give rise to poor performance for real videos due to degradation mismatch. Although some studies attempt to train deep models on noisy and noise-free video pairs captured by cameras, such models can only work well for specific cameras and do not generalize well for other videos. In this paper, we propose to lift this limitation and focus on the problem of **general real video denoising** with the aim to generalize well on unseen real-world videos. We tackle this problem by firstly investigating the common behaviors of video noises and observing two important characteristics: 1) downscaling helps to reduce the noise level in spatial space and 2) the information from the adjacent frames help to remove the noise of current frame in temporal space. Motivated by theses two observations, we propose a multi-scale recurrent architecture by making full use of the above two characteristics. Secondly, we propose a synthetic real noise degradation model by randomly shuffling different noise types to train the denoising model. With a synthesized and enriched degradation space, our degradation model can help to bridge the distribution gap between training data and real-world data. Extensive experiments demonstrate that our proposed method achieves the state-of-the-art performance and better generalization ability than existing methods on both synthetic Gaussian denoising and practical real video denoising. The codes will be made publicly available.

1 INTRODUCTION

Video denoising, with the aim of reducing the noise from a video to recover a clean video, has drawn increasing attention in low-level computer vision community (Tassano et al., 2019; 2020; Vaksman et al., 2021a; Davy et al., 2018; Chan et al., 2022c; Lee et al., 2021; Maggioni et al., 2021; Huang et al., 2022). Compared with image denoising, video denoising remains large underexplored domain. With the advance of deep learning (Ren et al., 2021; Zheng et al., 2021; Zamir et al., 2021), deep neural networks (DNNs) (Vaksman et al., 2021a; Tassano et al., 2020; Sheth et al., 2021) have become the dominant approach for video denoising. To push the envelope of video denoising, existing DNNs-based methods mainly focus on two directions with the some assumptions.

Firstly, a line of studies (Tassano et al., 2019; 2020) assume noisy videos are the addition of white Gaussian noises (AWGN) to clean videos. These methods perform well when tested on videos with the same degradation setting. However, their performance would deteriorates significantly when tested on videos corrupted by other types of noises (*e.g.*, video compression noise and camera sensor noise) due to the noise distribution mismatch (Zhang et al., 2022). To handle these noises, it is impractical to train multiple models. Moreover, noises in real-world videos are even more complex. Nevertheless, it is fair and necessary to train with AWGN and evaluate the effectiveness of different denoising methods in this simplified setup as a start point.

Secondly, to relieve the degradation mismatch between synthetic training data and real videos, the other line of work (Claus & van Gemert, 2019) proposed to capture noisy-clean video pairs for training. However, the video capturing and alignment process is time-consuming and expensive, which limits the potential size of such datasets. Another important limitation is that the training data is often captured by one specific camera, the degradation distribution of which may differ far

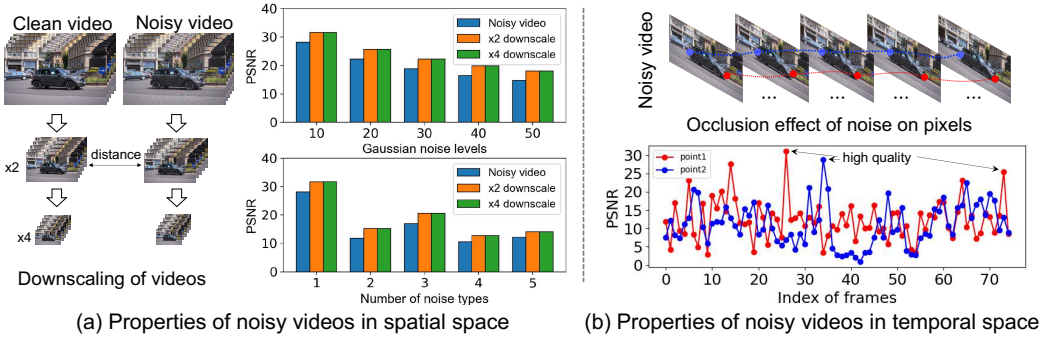


Figure 1: Two interesting properties of noisy videos. (a) Downscaling can remove part of noise in a video. (b) The pixels in a video have different degrees of occlusion. The high-quality adjacent pixels can help provide details to occluded pixels.

away from other cameras under other recording environments. Therefore, deep models (Claus & van Gemert, 2019) trained on such clean-noisy paired videos can suffer from poor generalization performance when tested on data collected from other cameras.

However, these two assumptions only consider limited types of degradations which rarely happen in real noisy videos. Such **degradation mismatch between training videos and real test videos** would inevitably give rise to poor generalization performance. To address this, we focus on a more general video denoising setup with the goal to train a deep model to generalize well to unseen real-world videos, different from existing studies illustrated in Figure 2. To tackle this problem, we first take a closer look on the inherent properties of noisy videos in the spatial and temporal space. The statistics of clean patches in noisy images have been explored in some studies (Zontak et al., 2013). However, there are little work devoted to the analysis of noisy videos. In Figure 1, we observe that downscaling can reduce part of noise for different levels. Motivated by this observation, we propose to integrate multi-scale learnable downscaling into the denoising network. On the other hand, noise in a video often has random patterns in temporal space. Some pixels in the current frame may have much more noise, while pixels in the same position of adjacent frames can have less noise, as shown in Figure 1 (b). To restore clean videos, it is necessary to model temporal connections so as to utilize information from adjacent frames to remove noise in the current frame.

Motivated by these two properties, we design a new architecture for general real video denoising, which we refer to as ReViD. Our network consists of multiple scales, each of which has learnable downscaling to remove spatial noise and recurrent modeling to separate temporal signal from a noisy video. To handle the degradation mismatch between training data and real-world test videos, we propose a new degradation model to generate diverse noisy videos and bridge the distribution gap by using a randomized composition of a wide range of degradations.

The contributions of this paper can be summarized as follows:

- We design a simple but effective real video denoising network by exploiting the inherent properties of a noisy video. Our method achieves the state-of-the-art performance on both additive white Gaussian denoising tasks and real-world video denoising tasks. Moreover, our model has faster runtime than Transformer-based methods.
- We make the first attempt for general real video denoising and propose a new noise degradation model. Our degradation model is able to generalize well on unseen and complex real-world videos. Moreover, we provide a theoretical analysis that training with our degradation model is equivalent to regularized loss with strong penalty. Our degradation model can generate diverse noisy video with large variance to better match the distribution of real-world videos.
- We conduct extensive experiments to demonstrate the effectiveness and superiority of our proposed method on both synthetic Gaussian denoising and practical real video denoising. We propose a new real-world video denoising test dataset consisting of different real-world noises. Our dataset can serve as a real video denoising benchmark for further studies.

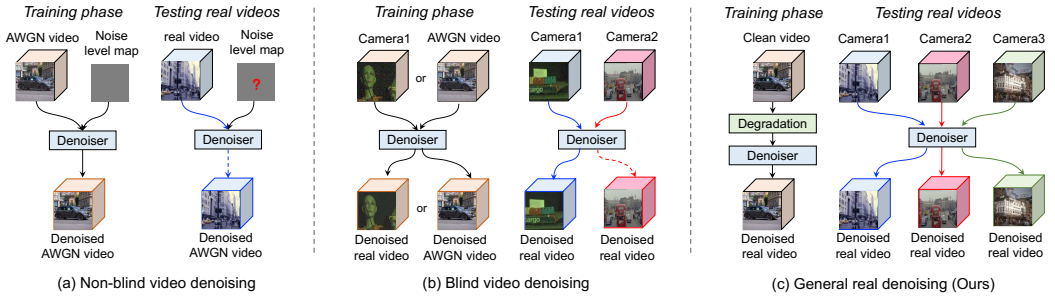
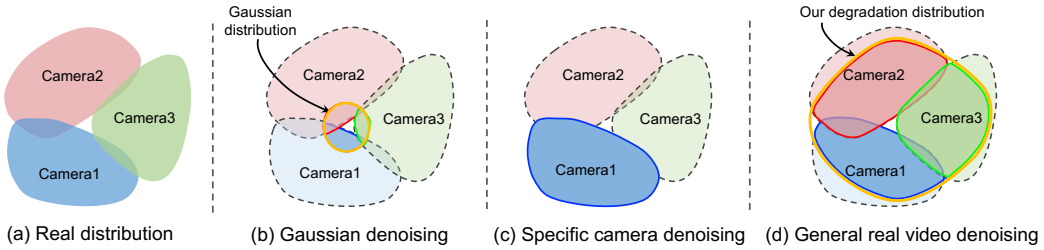


Figure 2: Discussion on the difference of existing video denoising setups. (a) Non-blind denoising methods take an AWGN video and its noise as input to synthesize a clean video. (b) Blind denoising methods aim to map a noisy video to a clean video without inputting the noise level. When training a model with noisy videos from a specific camera, it has poor performance (marked by the dotted line) on another camera. (c) Our general real denoising method first synthesizes different kinds of noisy videos with the degradation models, and then generalize well on different real-world videos.

2 Related Work

Image denoising aims to reduce noise from a noisy image (Kim & Ye, 2021; Fu et al., 2021; Luo et al., 2021; Bodrito et al., 2021). The well-known denoising algorithms BM3D (Dabov et al., 2007) and NLB (Lebrun et al., 2013) depend on the specific forms of prior and hand-tuned parameters in the optimization. They also lack flexibility as multiple models need to be trained for different levels of noise. To address this, recent methods exploit the benefits of deep neural networks to improve the image denoising. This includes convolution neural networks (CNNs) (*e.g.*, DnCNNs (Zhang et al., 2017), RBDN (Santhanam et al., 2017) and FFDNet (Zhang et al., 2018)) and Transformer (Liu et al., 2021) (*e.g.*, SwinIR (Liang et al., 2021) and SCUNet (Zhang et al., 2022)). In addition, many image denoising models (Plotz & Roth, 2017; Brooks et al., 2019) train on real image pairs captured by one cameras. However, these methods often have poor performance on other cameras. While image based denoising methods can in theory construct a baseline for real-world blind video denoising by treating each frame as a separate image, directly using them in our setup ignores the fruitful temporal connections between different frames in a video and leads to relatively poor performance.

Video denoising aims at removing noise to synthesize clean video sequences. Based on BM3D (Dabov et al., 2007), VBM4D (Maggioni et al., 2012) presents a video filtering algorithm to exploit temporal and spatial redundancy of a video. Some existing methods use the Recurrent Neural Network (RNN) to capture this sequential information. DRNNs (Chen et al., 2016) first applies deep RNN on the grady-scale images. However, this method seems to have difficulty to be extended to RGB images probably due to the difficulties of training RNN (Pascanu et al., 2013). Recently, BasicVSR++ (Chan et al., 2022a) improves the second-order grid propagation and flow-guided deformable alignment in RNN and extends video super-resolution to the video denoising (Chan et al., 2022c). In addition, some denoising methods adopt an asymmetric loss function (Vogels et al., 2018) to optimize the networks, or propose patch-based video denoising algorithm (Arias & Morel, 2018; Davy et al., 2018) to exploit the correlations among patches. PaCNet (Vaksman et al., 2021b) combines a patch-based framework with CNN by augmenting video sequences with patch-craft frames and inputting them in a CNN. To further improve over patch-based methods, DVDnet (Tassano et al., 2019) proposes spatial and temporal denoising blocks and trains them separately. To boost the efficiency, FastDVDnet (Tassano et al., 2020) extends DVDnet (Tassano et al., 2019) by using two denoising steps in the architecture which composed of a modified multi-scale U-Net (Ronneberger et al., 2015). VRT (Liang et al., 2022) proposes a video restoration transformer with parallel frame prediction, and achieves the state-of-the-art performance in video denoising. However, this transformer-based method has a large model size and expensive computational cost. Moreover, the above methods cannot be directly used in our real-world video denoising setup as they only consider synthesized Gaussian noise. Recently, ViDeNN (Claus & van Gemert, 2019) proposes a blind video denoising method trained either on AWGN noise or on collected real-world videos. However, this method may have limited generalization ability as the training only considers the specific noise type presented in the training dataset. This can lead to potential issues when tested on different real-wold videos captured from different sensors under different conditions.



(a) Real distribution (b) Gaussian denoising (c) Specific camera denoising (d) General real video denoising

Figure 3: Illustration of the degradation mismatch in different setups. The highlighted part indicates the training distribution. The dotted area illustrates mismatched test distribution that were unseen during training. (a) Real test distribution of videos captured from different cameras. (b) Training with Gaussian distribution cannot generalize well to a large area of the real distributions. (c) The model trained with collected dataset from one camera has poor performance on other cameras. (d) Our noise degradation aims to synthesize realistic videos with diverse noises to match the real distribution.

3 Proposed Method

3.1 General Real Video Denoising

In digital video processing, a noisy video can be corrupted by some random process. Formally, given a clean video sequence $\boldsymbol{x}$, a noisy video $\boldsymbol{x}_\sigma$ can be obtained by additive noises, *i.e.*, $\boldsymbol{x}_\sigma = \boldsymbol{x} + \boldsymbol{z}_\sigma$, where $\boldsymbol{z}_\sigma$ is a variable sampled from some distribution with density $p(\sigma)$. For traditional Gaussian denoising, this distribution is a zero-mean Gaussian distribution with standard deviation σ, *i.e.*, $\mathcal{N}(\mathbf{0}, \sigma \boldsymbol{I})$, where σ represents the noise level in a video. In addition, the noise distribution can come from the specific camera. However, real-world video noises are mostly unknown and can differ between different videos due to differences in unknown cameras, imaging setups, environments, *etc*. To improve the denoising performance on videos with unknown noises, we generalize the assumption on noises and do not assume any pre-defined noise type. We call this new setup Practical Real Video Denoising.

Practical real video denoising. As shown in Figure 2, unlike previous blind video denoising methods (Claus & van Gemert, 2019) which implicitly assume that the training and test data share the same noises, our proposed setup is more generalizable and can be tested on videos with unknown noises. Formally, our goal is to learn a video denoiser f to reduce noise and synthesize clean video sequence by minimizing the following problem, *i.e.*,

$$\widehat{f} = \arg\min_f \mathcal{L}(f) := \mathbb{E}_\sigma \left[\mathbb{E}_{\boldsymbol{x}} \left[\| f(\boldsymbol{x}_\sigma) - \boldsymbol{x} \|^2 \right] \right], \tag{1}$$

where $\mathbb{E}[\cdot]$ is an expectation *w.r.t.* the data or the noise distribution.

Degradation mismatching issue. Different real-world videos may come from different cameras and are processed by different ISP pipelines Brooks et al. (2019). Most existing methods suffer from degradation mismatching issue, *i.e.*, the distribution of synthesized training dataset mismatches the real videos, as shown in Figure 3. For traditional Gaussian denoising methods Tassano et al. (2019; 2020), Gaussian distribution and the real noise distribution rarely overlap, and thus methods have poor performance on real videos. Some other denoising methods Plotz & Roth (2017); Brooks et al. (2019); Claus & van Gemert (2019) train with degraded data from a specific camera, resulting in a degraded mismatch with the real video distribution from other cameras. To relieve the degradation mismatching issue, we first show how video noise properties can be exploited for network design to facilitate the optimization. We then propose a video degradation model to make the distributions of training data match better with real test videos.

3.2 Multi-Scale Recurrent Network for Video Denoising

In this section, we show how common properties of video noises can benefit network design in spatial and temporal video denoising. The proposed architecture is provided in Figure 4.

Denoising in the spatial space. As shown in Figure 1 (a), simple downscaling (*e.g.*, bicubic) can suppress specific noises (*e.g.*, Gaussian noise). However, simple downscaling is hard to handle more complex noises (*e.g.*, combination of different kinds of noises) in real-world videos and can also induce the serious blur artifacts. Therefore, we introduce a learnable convolution to downscale features to reduce different kinds of noise. Specifically, given an n-frame noisy video $\boldsymbol{x}_\sigma$, we first deploy a convolutional layer to extract low-level features $\{\widehat{\boldsymbol{g}}_1, \ldots, \widehat{\boldsymbol{g}}_n\}$. Here, $\boldsymbol{x}_\sigma$ is an input image

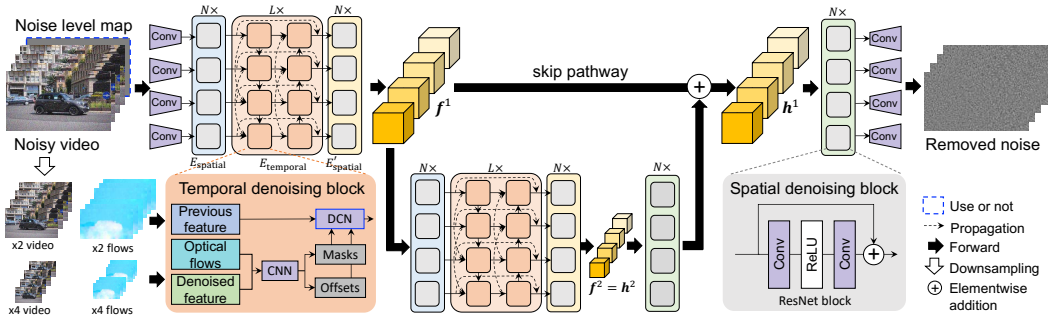


Figure 4: The architecture of the proposed multi-scale recurrent network. Our network is motivated by video noise properties. For non-blind video denoising, we take the noisy video and noise level map as an input. For practical real video denoising, we feed the noisy video augmented by our degradation models to train the network. At each scale, the network first removes spatial noise with learnable ResNet downscaling blocks and then removes temporal noise using a recurrent structure.

which combines the noisy video and the level map of the additive white Gaussian noise (AWGN) for traditional denoising problem. For real video denoising, $\boldsymbol{x}_\sigma$ is augmented by our proposed noise degradation model which is discussed in the next section. Then, we use a spatial encoder E_{spatial} to extract deep features and reduce the noise in space, *i.e.*,

$$\boldsymbol{g}_i^s = E_{\text{spatial}}\left(\boldsymbol{g}_i^{s-1}\right), \tag{2}$$

where $\boldsymbol{g}_i^0 = \widehat{\boldsymbol{g}}_i$, and the spatial encoder E_{spatial} can be modelled by multi-layered residual blocks.

Denoising in the temporal space. Motivated by the temporal property and (Chan et al., 2022a), given a denoised spatial feature $\boldsymbol{g}_i^s$, we use the optical-flow-guided deformable alignment as our temporal encoder E_{temporal} to compute the features

$$\begin{aligned}\widehat{\boldsymbol{f}}_{i,j}^s =& E_{\text{temporal}}\left(\boldsymbol{g}_i^s, \boldsymbol{f}_{i-1,j}^s, \boldsymbol{f}_{i-2,j}^s, \boldsymbol{o}_{i\to i-1}^s, \boldsymbol{o}_{i\to i-2}^s\right), \\ =& E_{dcn}([\boldsymbol{f}_{i-1}^s; \boldsymbol{f}_{i-2}^s], [\widetilde{\boldsymbol{o}}_{i\to i-1}^s; \widetilde{\boldsymbol{o}}_{i\to i-2}^s], [\boldsymbol{m}_{i\to i-1}^s; \boldsymbol{m}_{i\to i-2}^s]),\end{aligned} \tag{3}$$

where $\boldsymbol{f}_{i,j}^s$ is the feature at the i-th timestep in the j-th propagation branch at the s-th scale, and $\boldsymbol{o}_{i_1\to i_2}^s$ is the optical flow from i_1-th frame to the i_2-th frame, E_{dcn} is deformable convolution (DCN) (Zhu et al., 2019), $\widetilde{\boldsymbol{o}}_{i\to i-p}^s$ and $\boldsymbol{m}_{i\to i-p}^s$ are the offsets and masks which are formulated as

$$\widetilde{\boldsymbol{o}}_{i\to i-p}^s = \boldsymbol{o}_{i\to i-p}^s + c_1([\boldsymbol{g}_i^s; \bar{\boldsymbol{f}}_{i-1}^s; \bar{\boldsymbol{f}}_{i-2}^s]), \quad \boldsymbol{m}_{i\to i-p}^s = \tau(c_2([\boldsymbol{g}_i^s; \bar{\boldsymbol{f}}_{i-1}^s; \bar{\boldsymbol{f}}_{i-2}^s])), \tag{4}$$

where $p = 1, 2$, τ is the Sigmoid function, c_1 and c_2 are convolutional layers, and $\boldsymbol{f}_{i-1}^s$ is a warped feature using the optical flow $\boldsymbol{o}_{i\to i-1}^s$, *i.e.*, $\bar{\boldsymbol{f}}_{i-1}^s = \omega(\boldsymbol{f}_{i-1}^s, \boldsymbol{o}_{i\to i-1}^s)$ and $\bar{\boldsymbol{f}}_{i-2}^s = \omega(\boldsymbol{f}_{i-2}^s, \boldsymbol{o}_{i\to i-2}^s)$, where $\omega(\cdot)$ is a warp function according to the optical flow. After reducing the temporal noise, we use another spatial encoder E'_{spatial} to further remove the noise in space, *i.e.*,

$$\boldsymbol{f}_{i,j}^s = \widehat{\boldsymbol{f}}_{i,j}^s + E'_{\text{spatial}}\left(\left[\boldsymbol{f}_{i,j-1}^s; \widehat{\boldsymbol{f}}_{i,j}^s\right]\right), \tag{5}$$

where $[\cdot;\cdot]$ is a concatenation along the channel dimension and $\boldsymbol{f}_{i,0}^s = \boldsymbol{g}_i^s$. Let $\boldsymbol{f}_i^s$ be the feature in the last branch at the s-th scale, the spatial decoder D_{spatial} aggregates features with the skip connection,

$$\boldsymbol{h}_i^s = \boldsymbol{f}_i^s + D_{\text{spatial}}\left(\boldsymbol{h}_i^{s+1}\right), \tag{6}$$

where $\boldsymbol{h}_i^S = \boldsymbol{f}_i^S$ at the last scale S and spatial decoder can be implemented by multi-layered residual blocks (He et al., 2016) with PixelShuffle (Shi et al., 2016). Last, we use convolutional layers to produce residual noise. In the training, we first train a denoiser using L1 loss, and then we further train the model by minimizing a weighted combination of L1 loss, perceptual loss and GAN loss.

Difference from BasicVSR++. Our architecture design differs from BasicVSR++ in the following aspects. First, our denoiser is built on the U-Net architecture with downscaling and upscaling, which is effective to capture spatio-temporal information for video denoising. Specifically, in downscaling, features are extracted at different scales by both spatial denoising and temporal propagation. Multi-scale optical flows are used for guidance in alignment, so as to deal with different motion magnitudes. In upscaling, we only do spatial modelling to save computation cost. In contrast, BasicVSR++ does not use multiscale modelling which is important in video denosing as shown in Figure 1(a). Second, BasicVSR++ directly downsamples the inputs using Bicubic interpolation. Such downsampling can remove part of noise but also remove some useful texture information. In contrast, we propose to train with learnable parameters to remove noise and preserve the useful texture information. Since BasicVSR++ is designed for video super-resolution rather than video denoising, directly applying it for video denoising would result in inferior performance.

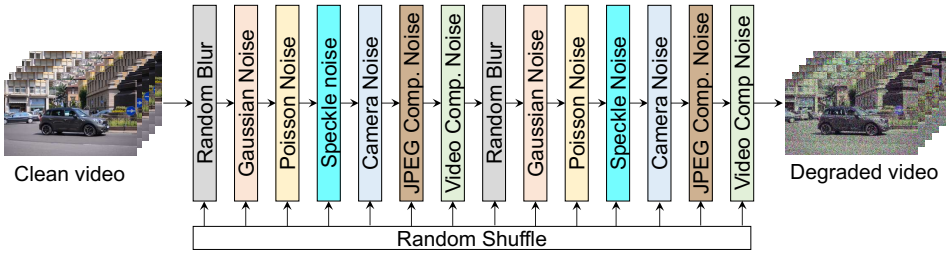


Figure 5: An illustration of the proposed noise degradation pipeline. For a high quality video, a randomly shuffled degradation sequence is performed to produce a noisy video.

3.3 Real Noise Degradations

Unlike Gaussian noises in traditional setups, real-world videos often contain unknown noises and blur and they differ from video to video. To jointly remove noises and blur, we train our model with general video degradations. To cover a large range of real video distribution, we randomly change the order of different degradations, and augment the data with these degradations. Formally, given a clean video $\boldsymbol{x}$, we use composition function of N shuffled degradations to synthesize a noisy video:

$$\boldsymbol{x}_\sigma = g(\boldsymbol{x}) = \left(g_{i_1} \circ g_{i_2} \circ \cdots \circ g_{i_N}\right)(\boldsymbol{x}), \quad \text{where} \quad \{i_1, \ldots, i_N\} = \phi(\{1, \ldots, N\}), \tag{7}$$

where ϕ is a shuffle function, $\circ$ is a function composition, and g_{i_n} is a degradation model of the i_n-th type. Motivated by (Bishop, 1995), we prove the following theorem to understand our degradation.

Theorem 1 (Effect of noise degradations) *Let $\boldsymbol{z}_\sigma = g(\boldsymbol{x}) - \boldsymbol{x}$, and assume that the mean and variance of the noise distribution are 0 and $\eta^2(\boldsymbol{z}_\sigma)$, then the loss (1) , i.e.,*

$$\mathbb{E}_\sigma\left[\mathbb{E}_{\boldsymbol{x}}\left[\|f(\boldsymbol{x}_\sigma) - \boldsymbol{x}\|^2\right]\right] = \mathbb{E}_{\boldsymbol{x}}\left[\|f(\boldsymbol{x}) - \boldsymbol{x}\|^2\right] + \eta^2(\boldsymbol{z}_\sigma)\mathbb{E}_{\boldsymbol{x}}\left[\left\|\frac{\partial f}{\partial \boldsymbol{x}}\right\|^2 + \frac{1}{2}\left(f(\boldsymbol{x}) - \boldsymbol{x}\right)^\top \frac{\partial^2 f}{\partial \boldsymbol{x}^2}\mathbf{1}\right]. \tag{8}$$

From this theorem, the loss (1) trained with our noise degradations is equivalent to a normal loss with a regularization term. The parameter $\eta^2(\boldsymbol{z}_\sigma)$ is related to the amplitude or variance of the noise $\boldsymbol{z}_\sigma$ and controls how the regularization term influences the loss. Moreover, our degradation model make $\eta^2(\boldsymbol{z}_\sigma)$ to be large (see Figure 11) to improve the generalization performance of our model.

Noise. Noises in real-world videos come from different sources. To simulate such noises, we propose noise degradations, including Gaussian noise, Poisson noise, Speckle noise, Processed camera sensor noise, JPEG compression noise and video compression noise.

- *Gaussian noise.* When there are no prior information of noise, one can add Gaussian noise into a video sequence. Given a clean video $\boldsymbol{x}$, we synthesize a noisy video $g_1(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{z}$, where the noise $\boldsymbol{z}$ can be additive white Gaussian noise (AWGN) and gray-scale AWGN.
- *Poisson noise.* In electronics, Poisson noise is a type of shot noise which occurs in photon counting in optical devices. Such noise arises from the discrete nature of electric charge, and it can be modeled by a Poisson process. Given a clean video $\boldsymbol{x}$, we synthesize a noisy video by $g_2(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{z}$, where $\boldsymbol{z} = \boldsymbol{z}' - \boldsymbol{x}$ and $\boldsymbol{z}' \sim \mathcal{P}(10^\alpha \cdot \boldsymbol{x})/10^\alpha$.
- *Speckle noise.* Speckle noise exists in the synthetic aperture radar (SAR), medical ultrasound and optical coherence tomography images. We simulate such noise by multiplying the clean image $\boldsymbol{x}$ and Gaussian noise $\boldsymbol{z}$, *i.e.*, $\boldsymbol{x} * \boldsymbol{z}$. Then, we synthesize noisy video by $g_3(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{x} * \boldsymbol{z}$.
- *Processed camera sensor noise.* In modern digital cameras, the processed camera sensor noise originates from the image signal processing (ISP). Inspired by (Zhang et al., 2022), the reverse ISP pipeline first get the raw image from an RGB image, then the forward pipeline constructs noisy raw image by adding noise to the raw image, which denoted by $g_4(\boldsymbol{x}) = \text{forward}(\text{reverse}(\boldsymbol{x}))$.
- *JPEG compression noise.* It is widely used to reduce the storage for digital images with the fast encoding and decoding (Zhang et al., 2021). We synthesize frames with JPEG compression noise by $g_5(\boldsymbol{x}) = \text{Dec}(\text{Enc}(\boldsymbol{x}))$. Such JPEG compression methods often cause 8×8 blocking artifacts.
- *Video compression noise.* Videos sometimes have compression artifact and presents on videos encoded in different format. We use the Pythonic operator av in FFmpeg, *i.e.*, $g_6(\boldsymbol{x})$=av$(\boldsymbol{x})$.

Apart from noise, most real-world videos inherently suffer from bluriness. Thus, we additionally consider two common blur degradations, including Gaussian blur and resizing blur. For Gaussian blur, we synthesize a video as $g_7(\boldsymbol{x}) = \mathbf{x} * \boldsymbol{\kappa}$, where '$*$' is the convolution operator and $\boldsymbol{\kappa}$ is the Gaussian kernel. For resizing blur, we first downscale a video for $s\times$ and then upscale to the original size, *i.e.*, $g_8(\boldsymbol{x}) = \text{up}_s(\text{down}_{\frac{1}{s}}(\boldsymbol{x}))$, where $\text{down}_{\frac{1}{s}}$ and up_s are downscaling and upscaling function.

Table 1: Quantitative comparison (average RGB channel PSNR) with state-of-the-art methods for video denoising on the DAVIS and Set8 datasets. Best results are in **bold**.

Datasets	σ	VBM4D	VNLB	DVDnet	FastDVDnet	VNLNet	PaCNet	BasicVSR++	VRT	**ReViD (Ours)**
DAVIS	10	37.58	38.85	38.13	38.71	39.56	39.97	40.13	40.82	**41.03**
	20	33.88	35.68	35.70	35.77	36.53	36.82	37.41	38.15	**38.50**
	30	31.65	33.73	34.08	34.04	-	34.79	35.74	36.52	**36.97**
	40	30.05	32.32	32.86	32.82	33.32	33.34	34.49	35.32	**35.83**
	50	28.80	31.13	31.85	31.86	-	32.20	33.45	34.36	**34.90**
Set8	10	36.05	37.26	36.08	36.44	37.28	37.06	36.83	37.88	**38.07**
	20	32.18	33.72	33.49	33.43	34.08	33.94	34.15	35.02	**35.35**
	30	30.00	31.74	31.79	31.68	-	32.05	32.57	33.35	**33.78**
	40	28.48	30.39	30.55	30.46	30.72	30.70	31.42	32.15	**32.66**
	50	27.33	29.24	29.56	29.53	-	29.66	30.49	31.22	**31.77**
Params. (M)		-	-	**0.48**	2.48	-	2.87	9.76	18.3	13.68
Runtime (s)		420.0	156.0	2.51	**0.08**	1.65	35.24	**0.08**	5.91	0.32

Table 2: Quantitative comparison in PSNR for denoising clipped Gaussian noise on DAVIS.

Methods	Noise levels			Average
	10	30	50	
ViDeNN	37.13	32.24	29.77	33.05
FastDVDnet	38.65	33.59	31.28	34.51
PaCNet	39.96	34.66	32.00	35.54
ReViD-blind	40.94	36.79	34.65	37.46
ReViD	**41.00**	**36.91**	**34.83**	**37.58**

Table 3: Quantitative comparison in PSNR for single image denoising on Set8 dataset.

Methods	Noise levels			Average
	15	25	50	
BM3D	29.00	28.64	26.50	28.05
Restormer	34.36	31.40	28.57	31.44
SwinIR	34.87	32.37	29.19	32.14
SCUNet	34.82	32.34	29.14	32.10
ReViD	**36.47**	**34.49**	**31.77**	**34.24**

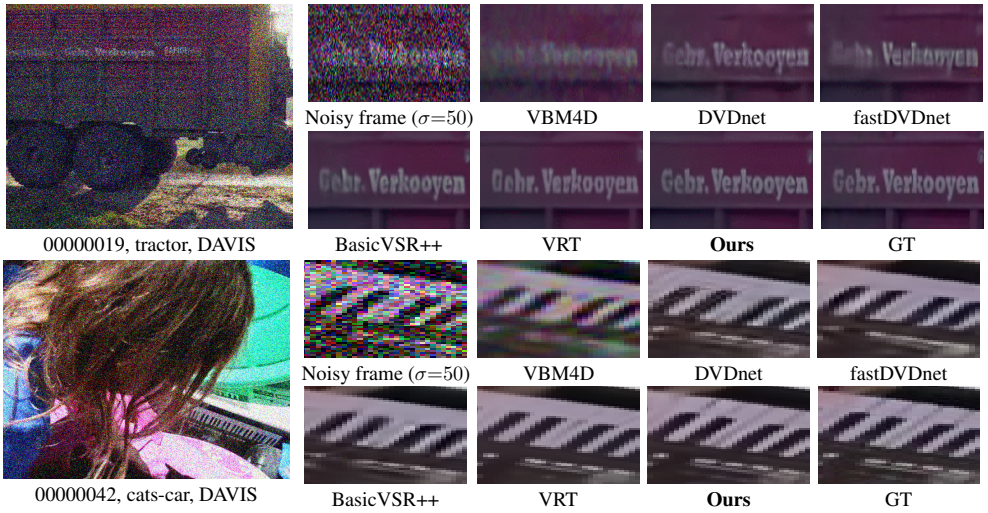


Figure 6: Visual comparison of different methods on DAVIS under the noise level of 50.

4 Experiments

4.1 Synthetic Gaussian Denoising

Datasets. We use DAVIS (Khoreva et al., 2018) and Set8 (Tassano et al., 2019) in synthetic Gaussian denoising. Following the setting of (Liang et al., 2022), we synthesize the noisy video sequences by adding AWGN with noise level $\sigma \in [0, 50]$ on the DAVIS (Khoreva et al., 2018) training set. We then train the model by using the synthesized data and test it on the DAVIS testing set and Set8 (Tassano et al., 2019) with different Gaussian noise levels $\{10, 20, 30, 40, 50\}$.

Quantitative comparison. Tables 1-3 show PSNR (Chan et al., 2022a) between different methods on the test datasets DAVIS (Khoreva et al., 2018) and Set8 (Tassano et al., 2019). Our method has best performance on both DAVIS and Set8 with a large margin. Specifically, our model outperforms BasicVSR++ (Chan et al., 2022c) by an average PSNR of **1.23db**. Moreover, we also train a blind model for clipped AWGN to obtain the best performance. In Figure 7, our model achieves the best performance gains with similar model size and runtime. In particular, for the largest noise level of 50, our model outperforms VRT (Liang et al., 2022) with a smaller model size and faster inference time. Our model yields a PSNR improvement of **0.54db**.

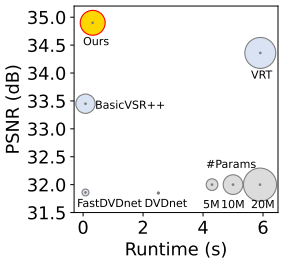


Figure 7: Runtime, PSNR, and model size.

Table 4: Quantitative Comparison of different methods on VideoLQ and NoisyCity4 for the practical video denoising task. For fair comparison, we train BasicVSR++ and RealBasicVSR on the same proposed noise degradation pipeline, which is denoted by suffix '*'.

Methods	VideoLQ			NoisyCity4		
	NIQE↓	BRISQUE↓	PIQE↓	NIQE↓	BRISQUE↓	PIQE↓
SCUNet	4.7797	39.6360	68.7677	5.1971	51.5672	85.2371
Restormer	4.3755	39.9023	69.6296	5.1884	52.7126	86.2248
ViDeNN	4.2722	33.8539	60.7876	4.7613	42.5865	78.9111
BasicVSR++	4.0233	34.9458	51.4780	5.4899	52.1469	81.1234
BasicVSR++*	4.2879	29.1541	49.1658	4.4235	33.4198	**47.5131**
RealBasicVSR*	4.2167	29.2103	48.0369	4.0578	26.3504	51.5825
Ours-real	**4.0205**	**29.0212**	**45.0768**	**3.8540**	**24.2025**	48.2962

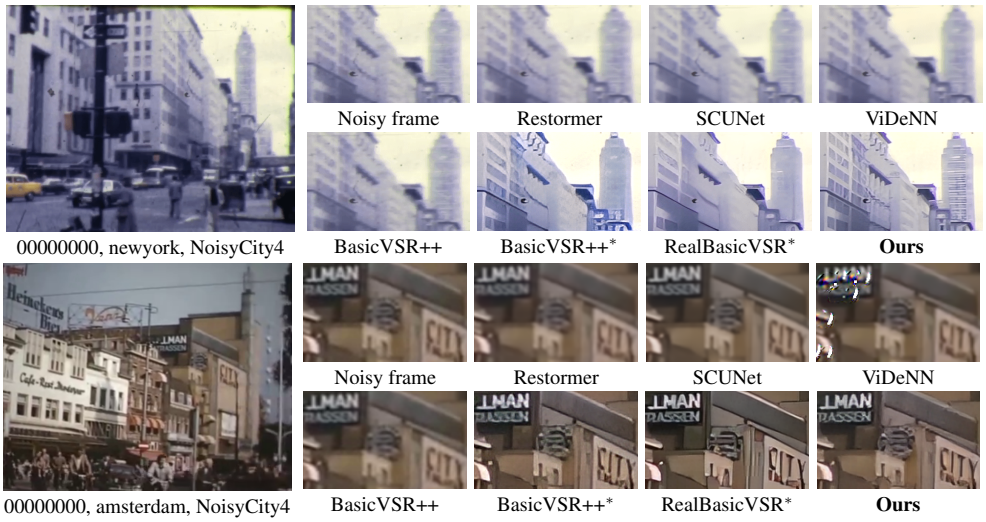


Figure 8: Visual comparison of different video denoising methods on NoisyCity4.

Qualitative comparison. In Figure 6, we compare different video denoising methods under the high noise level of 50. Our proposed denoiser restores better structures and preserves clean edge than previous state-of-the-art video denoising methods, even though the noise level is high. In particular, our model is able to restore the letters 'Gebr' in the first example and piano texture in the second example of Figure 6. In contrast, VBM4D (Maggioni et al., 2012), DVDnet (Tassano et al., 2019) and FastDVDnet (Tassano et al., 2020) fail to remove severe noise from a video frame. BasicVSR++ (Chan et al., 2022a) and VRT (Liang et al., 2022) only restore part of the textures.

4.2 Practical Real Video Denoising

Figure 9: Examples of the NoisyCity4 dataset.

For real video denoising, we use REDS (Nah et al., 2019) as the training set. According to the setting of (Wang et al., 2019), we use 266 regrouped training clips in REDS (Nah et al., 2019), where each with 100 consecutive frames. Specifically, we synthesize noisy video sequences on the REDS training set by using our proposed noise degradation model. To evaluate the generalizability of real-world video denoising methods, one can use VideoLQ (Chan et al., 2022b) which is downloaded from Flickr and YouTube and contains 50 video sequences, where each with up to 100 frames. However, the VideoLQ dataset was mainly proposed for real-world video super-resolution and it has low level of noise itself. To address this, we additionally propose a new benchmark dataset for real-world video denoising, called **NoisyCity4** dataset. This dataset is collected from YouTube and contains four city street videos from decades ago. The videos in the proposed dataset contain real-world noises from different sources such as film grains, film scratches, flickers etc. Examples of the NoisyCity4 videos are shown in Figure 9 and further provided in the supplementary material. Each video in NoisyCity4 contains a sequence of 100 frames with different noises.

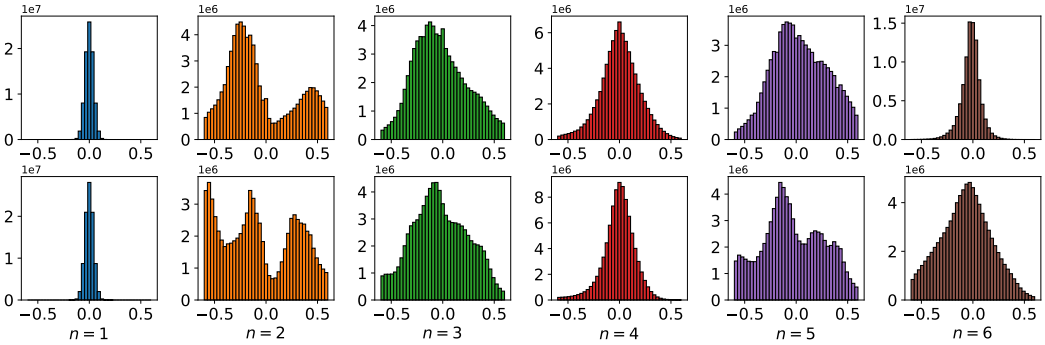


Figure 10: Distributions of noise degradation without (Top) and with (Bottom) random shuffle.

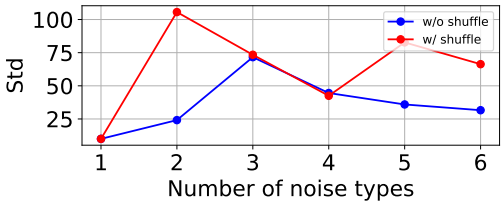


Figure 11: Variance of noise degradations.

Table 5: Ablation study of our model on DAVIS and Set8 in PSNR under the noise level of 50.

Methods	DAVIS	Set8
w/o spatial module	34.45	31.12
w/o temporal module	31.90	29.59
downscaling three times	34.91	31.75
ReViD (ours)	**34.90**	**31.77**

Quantitative comparison. In Table 4, we compare different methods on VideoLQ (Chan et al., 2022b) and NoisyCity4. Here, we use three non-reference metrics NIQE (Mittal et al., 2012), BRISQUE (Mittal et al., 2011) and PIQE (Venkatanath et al., 2015) as evaluation metrics because they are commonly used to measure the quality of images and ground-truth videos are not available. Our model achieves better performance than all other methods under all metrics. In contrast, it is difficult for ViDeNN to reduce noise in real video since the videos are captured by different cameras. With the help of our noise degradation model, the denoisers are able to reduce the real-world noise.

Qualitative comparison. As shown in Figure 8, our model achieves the best visual quality among different methods. By taking the spatial and temporal properties into account and using the proposed noise degradation model, our denoiser improves visual quality and leads to cleaner details and edges than other methods. For instance, our model is able to recover the windows in the building. In contrast, it is hard for image based denoisers (Zamir et al., 2022; Zhang et al., 2022) and ViDeNN (Claus & van Gemert, 2019) to remove the noise well in a real-world video. There results demonstrate our degradation model is able to improve the generalization ability.

4.3 Further Experiments

Effect of our degradation model. To study the effect of our degradation model, we show the distributions of the synthesized noise by our degradation model with and without the proposed random shuffle in Figure 10. The random shuffle strategy can improve the diversity of the synthesized distributions. In addition, this strategy can increase the noise variance in Figure 11. This shows that the proposed method can generate more diverse distributions in the training.

Ablation study. We investigate the effectiveness of the spatial and temporal modules in Table 5. Specifically, we conduct experiments by removing these modules. The model without these modules has performance drop, which demonstrates the importance of them. In addition, we investigate the performance by increasing the times of downscaling to 3. The model has comparable PSNR but with larger model size. Thus, we downscale the videos twice in the experiment.

5 Conclusion

In this paper, we propose a practical and important setup in video denoising called practical real video denoising. Motivated by properties of video noises, we first propose a real video denoising network, called ReViD to achieve the state-of-the-art performance on synthetic Gaussian denoising and general real video denoising. Moreover, we make the first attempt to design a new noise degradation model for the real-world video denoising task which considers different kinds of noise with random shuffle. In addition, we propose a new real video denoising dataset with different levels of noise. Extensive experiments demonstrate the effectiveness and superiority of denoising and practicability of our method. Besides, our model has good generalization performance on unseen real videos.

REFERENCES

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

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 1995.

Théo Bodrito, Alexandre Zouaoui, Jocelyn Chanussot, and Julien Mairal. A trainable spectral-spatial sparse coding model for hyperspectral image restoration. *Advances in Neural Information Processing Systems*, 2021.

Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022a.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022b.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. On the generalization of basicvsr++ to video deblurring and denoising. *arXiv preprint arXiv:2204.05308*, 2022c.

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

Michele Claus and Jan van Gemert. Videnn: Deep blind video denoising. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 2007.

Axel Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. Non-local video denoising by cnn. *arXiv preprint arXiv:1811.12758*, 2018.

Xueyang Fu, Zeyu Xiao, Gang Yang, Aiping Liu, Zhiwei Xiong, et al. Unfolding taylor's approximations for image restoration. *Advances in Neural Information Processing Systems*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, 2016.

Cong Huang, Jiahao Li, Bin Li, Dong Liu, and Yan Lu. Neural compression-based feature learning for video restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision*, 2018.

Kwanyoung Kim and Jong Chul Ye. Noise2score: Tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 2021.

Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 2013.

Seunghwan Lee, Donghyeon Cho, Jiwon Kim, and Tae Hyun Kim. Restore from restored: Video restoration with pseudo clean video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *IEEE International Conference on Computer Vision Workshops*, 2021.

Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision*, 2021.

Fangzhou Luo, Xiaolin Wu, and Yanhui Guo. Functional neural networks for parametric image restoration problems. *Advances in Neural Information Processing Systems*, 2021.

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*, 2012.

Matteo Maggioni, Yibin Huang, Cheng Li, Shuai Xiao, Zhongqian Fu, and Fenglong Song. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Anish Mittal, Anush K Moorthy, and Alan C Bovik. Blind/referenceless image spatial quality evaluator. In *Asilomar Conference on Signals, Systems and Computers*, 2011.

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 2012.

Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 2013.

Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *IEEE conference on computer vision and pattern recognition*, 2017.

Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. Adaptive consistency prior based deep network for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

Venkataraman Santhanam, Vlad I Morariu, and Larry S Davis. Generalized deep image to image regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Dev Yashpal Sheth, Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter A Crozier, Mitesh M Khapra, Eero P Simoncelli, and Carlos Fernandez-Granda. Unsupervised deep video denoising. In *IEEE International Conference on Computer Vision*, 2021.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, 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 conference on computer vision and pattern recognition*, 2016.

Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *IEEE International Conference on Image Processing*, 2019.

Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Gregory Vaksman, Michael Elad, and Peyman Milanfar. Patch craft: Video denoising by deep modeling and patch matching. In *IEEE International Conference on Computer Vision*, 2021a.

Gregory Vaksman, Michael Elad, and Peyman Milanfar. Patch craft: Video denoising by deep modeling and patch matching. In *IEEE International Conference on Computer Vision*, 2021b.

N Venkatanath, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya, and Swarup S Medasani. Blind image quality evaluation using perception based features. In *National Conference on Communications*, 2015.

Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics*, 2018.

Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

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*, 2017.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 2018.

Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE International Conference on Computer Vision*, 2021.

Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Radu Timofte, and Luc Van Gool. Practical blind denoising via swin-conv-unet and data synthesis. *arXiv preprint arXiv:2203.13278*, 2022.

Hongyi Zheng, Hongwei Yong, and Lei Zhang. Deep convolutional dictionary learning for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Maria Zontak, Inbar Mosseri, and Michal Irani. Separating signal from noise using patch recurrence across scales. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.