Supplementary Material to Neural Compression-Based Feature Learning for Video Restoration

Cong Huang[1*] Jiahao Li [2] Bin Li[2] Dong Liu[1] Yan Lu[2]
[1] University of Science and Technology of China [2] Microsoft Research Asia
hcy96@mail.ustc.edu.cn, dongeliu@ustc.edu.cn, {li.jiahao, libin, yanlu}@microsoft.com

1. Overview

This document provides the supplementary material to our proposed neural compression-based feature learning for video restoration, including detailed network architecture, training details for different tasks, and additional ablation studies. More comparisons on video denoising, video deraining and video dehazing are also presented. We will release our source code upon acceptance. We also provide the restored videos to verify the authenticity of our scheme.

2. Network Architecture

Our framework contains three parts: feature alignment, feature refinement (including feature attention module and neural compression-based feature learning module), and feature fusion. In this supplementary material, we present the network architecture details.

MV refinement module. The structure of the MV refinement module is shown in Fig. 1 (a). The MV refinement module encodes the corrupted MV into a compact representation and then decodes it to the refined MV. Specifically, we use two convolutional layers with stride=2 to encode the corrupted MV and two deconv layers to decode the refined MV. The number of the channel of the intermediate features is 64. As GDN [1] could reduce the statistical dependencies of the features and compact the features, we use GDN [1] in the encoder and use the inverse-GDN (IGDN) in the decoder correspondingly.

Feature attention module. As Fig. 1 (b) shows, the feature attention module takes the noisy frame x_t and aligned features $\hat{c}_t$ as input, then generates the spatial-channel attention map m_t. The feature attention module is based on an auto-encoder with the following modifications. First, we use Resblock [11] to extract the features at each scale. Second, we add 4 Resblocks at the largest scale to increase the receptive field with minor computation cost. To keep the feature attention module lightweight, the numbers of the channels of the intermediate features are 16, 32, 64 for three scales, respectively.

Neural compression-based feature learning module. As Fig. 2 shows, the neural compression-based feature learning module contains a prior model, a feature encoder, and a feature decoder. The prior model learns to estimate the parameters (μ_t, σ_t, q_t) that are used in the adaptive quantization and the $\mathcal{L}oss_{CE}$. The feature encoder encodes the features $\check{c}_t$ into compact latent codes e_t. Then the latent codes e_t are processed by our proposed adaptive quantization. At last, the refined features $\tilde{c}_t$ are decoded from the processed latent codes $\hat{e}_t$. For the prior model, we use three convolutional layers (two layers are with stride=2) to estimate the parameters, where LeakyReLU is used as the activation function. For the feature encoder and decoder, we use two convolutional layers with stride=2 to encode the features and two deconvolutional layers to decode the noise-robust features. The number of the channel of the intermediate features is 64.

Restoration module. As Fig. 3 shows, the restoration module fuses the noise-robust features $\tilde{c}_t$ with the current frame x_t, and then generates the final output frame $\tilde{y}_t$. At the same time, the features c_t used by next step are also generated. Following previous image restoration methods [5, 8], we adopt a deep U-Net [15] as our restoration module for synthetic video denoising, video deraining and video dehazing. We also leverage the channel-attention block [25] (CAB) to extract the features at each scale. The structure of CAB is shown in Fig. 3 (b) and the GAP means the global average pooling. The features from the skip connection are processed by a convolutional layer. The numbers of the channels of the intermediate features are 32, 64, 128, 256, 512 for five scales, respectively. We also propose a more powerful W-Net-like [20] restoration module for real-world video denoising which is a more challenging task. As Fig. 4 shows, we cascade two U-Net restoration modules as our W-Net-like restoration module. The number of the channel of the intermediate features of each scale remains unchanged.

*This work was done when Cong Huang was an intern at Microsoft Research Asia.

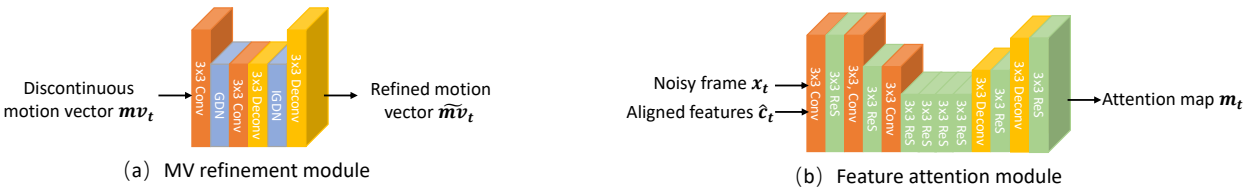

Figure 1. The structure of MV refinement module and feature attention module.

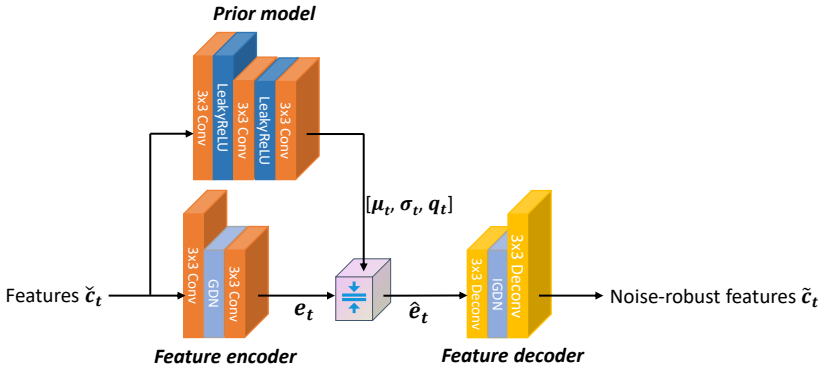

Figure 2. The structure of neural compression-based feature learning module.

3. Training Details

We adopt AdamW optimizer and cosine annealing learning rate scheduler. The initial learning rates of the motion estimation module and other modules are set to 2.5e-5 and 2e-4, respectively. The weights of the motion estimation module are fixed during the first 2500 iterations. The batch size is 16 and each sample in the batch is a 5-frame video clip. The patch size is 128x128. The data augmentation includes random horizontal, vertical, and transposed flipping.

During the training, we use a two-stage training scheme to learn the noise-robust feature representation and then make this feature representation help the final reconstruction. In the first stage, we use the $\mathcal{L}oss_{\mathcal{L}2}$ and $\mathcal{L}oss_{CE}$ to train the model for 50k iterations. The first stage training not only provides the feature generation a good starting point but also helps the compression module converge to a relatively stable status which can effectively filter the noisy and irrelevant information. The second stage only use $\mathcal{L}oss_{\mathcal{L}2}$ for the rest iterations. With a well-trained compression module, the second stage carefully fine-tunes the quantization step size and the mean value only guided by the $\mathcal{L}oss_{\mathcal{L}2}$. This helps the model focus more on the model's generation ability for better reconstruction quality. This simple two-stage training helps the temporal features be robust to the noise, and then lets these features improve the final quality.

The main difference of training setting among different datasets is the number of total iterations. For DAVIS [14] dataset and CRVD [23] dataset, we train our model for 100k iterations and 200k iterations, respectively. For RainSynLight25 [12], RainSynComplex25 [12], RainSynAll100 [22] and NTU-Rain [6], we train our model for 200k, 200k, 250k and 100k iterations, respectively. For REVIDE [24], we train our model for 50k iterations. Besides, we follow the setting in CG-IDN [24] and use patch size 384x384 in our method for REVIDE.

4. More Ablation Studies

We conduct more ablation studies about the effect of different modules, the weight of the cross-entropy loss, and the effect of the neural compression-based feature learning on clean features.

4.1. The Effect of Different Modules

This paper proposes three key modules: the MV refinement (MVR) , the neural compression-based feature learning (NCFL) incorporated with adaptive quantization, and the feature attention (FA). We study the effect of these modules and report the results in the main paper. In this supplementary material, we conduct the experiments by using normal convolutional layers (without auto-encoder structure) to replace MVR and NCFL, for demonstrating that the improvements of our modules are not from increasing complexity or deeper structure. MVR is replaced by normal convolutional layers, denoted as M-Conv. NCFL is replaced by normal convolutional layers, denoted as N-Conv. The

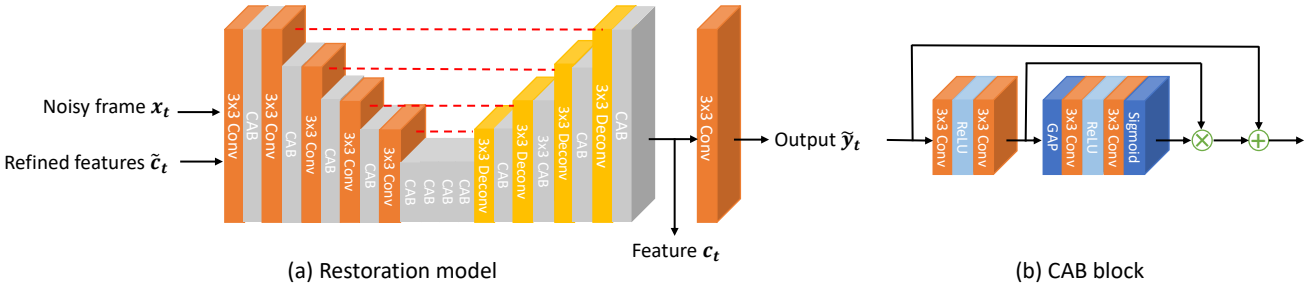

Figure 3. The structure of U-Net-like restoration module and CAB block.

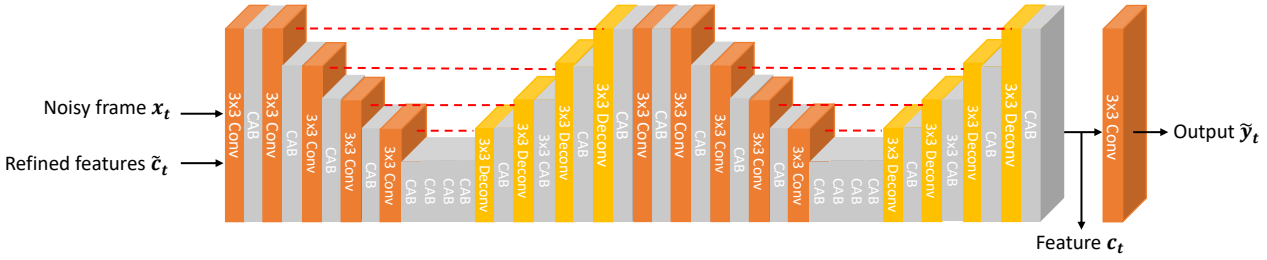

Figure 4. The structure of W-Net-like restoration module.

	M_a	M_b	M_c	M_d	M_e	M_f
MVR		✓		✓	✓	✓
M-Conv			✓			
NCFL				✓		✓
N-Conv					✓	
FA						✓
PSNR	29.75	29.87	29.79	30.29	29.93	30.45
GFLOPs	534	548	557	613	639	771

Table 1. The ablation study on different modules. Tested on Set8 ($\sigma = 50$). MVR denotes the MV refinement. NCFL denotes the neural compression-based feature learning. FA denotes the feature attention. M-Conv represents that we use the normal convolutional layers with slightly larger complexity than MVR to replace MVR. N-Conv represents that we use the normal convolutional layers with slightly larger complexity than NCFL to replace NCFL.

complexity of M-Conv and N-Conv is slightly higher than that of MVR and NCFL, respectively. As Table 1 shows, replacing MVR with M-Conv causes a 0.08 dB PSNR drop ($M_b \rightarrow M_c$) and replacing NCFL with N-Conv causes a 0.36 dB PSNR drop ($M_d \rightarrow M_e$). These results verify that the improvement of performance is not brought by the increasing complexity but comes from our special design.

λ	1/256	1/512	1/1024	1/2048	1/4096
PSNR	30.25	30.29	30.35	30.45	30.40

Table 2. The ablation study on the weight λ of $\mathcal{L}oss_{Entropy}$. Tested on Set8 ($\sigma = 50$).

4.2. The Weight of Cross-Entropy Loss

In the first stage training, the weight λ of the cross-entropy loss $\mathcal{L}oss_{CE}$ controls the intensity of filtering the noisy information in the temporal features. As Table 2 shows, if λ is large (e.g., λ = 1/256 or 1/512), some useful information may also be filtered, which degrades performance a bit. On the contrary, if λ is too small (e.g., λ = 1/4096), the temporal features may still contain some noisy information and performance also drops. The optimal λ may be related to the noise level, i.e. different noise levels require different λ to achieve the best performance. To avoid too many parameters tuning, we use 1/2048 as the default value of λ for other experiments. In the future, we will investigate how to adaptively set λ according to the noise level and content characteristic.

4.3. The Effect of Neural Compression-Based Feature Learning on Clean Features

Our neural compression-based feature learning is used to filter the noisy and irrelevant information in noisy features,

and it will not discard useful information in the clean features. To verify it, we use clean frames from DAVIS trainval dataset as input and groundtruth to train the individual NCFL module with adaptive quantization (NCFL-AdapQ) and the NCFL module without quantization (NCFL-NoQ). We do not train our complete model because we directly use the current input frame as the input of restoration module. This will enable the restoration module to directly learn the identity mapping when the input is clean, and make the temporal feature propagation useless. The results show that NCFL-AdapQ reaches PSNR 30.37 dB and NCFL-NoQ reaches PSNR 30.39 dB. The similar results of these two models show that the proposed adaptive quantization has little impact on the clean features.

5. More Comparisons on Video Denosing

We compare our method with these baselines: VNLNet [7], FastDVDNet [17], EMVD [13], EDVR [18], BasicVSR [2], and BasicVSR++ [3]. EMVD has several network structure configurations with different complexities. More specifically, EMVD mainly contains three modules: the fusion module, the denoising module, and the refinement module. The number of convolutional layers and the number of the channel of the intermediate features determine the complexity. The default setting of EMVD is (f-2-16, d-2-16, r-2-16). 'f-2-16' means that the fusion module contains 2 convolutional layers and the number of the channel of the intermediate feature is 16. 'd-2-16' represents the configuration for the denoising module and 'r-2-16' represents the configuration for the refinement module. We denote the default setting as EMVD-small (EMVD-S, f-2-16, d-2-16, r-2-16). In addition, we also compare another setting, i.e. EMVD-large (EDVR-L, f-4-64, d-6-256, r-4-64). The two settings can be found in the Table 1(b) in EMVD paper [13]. Since the official code of EMVD is not released, we use the third-party implementation [4].

Quantitative Comparison. In the main paper, we have shown the results on Set8 testset when compared with previous neural network-based methods. In addition, to help us better understand the advantage of neural network-based method over traditional method, we also test median filtering and summarize the comparison in Table 5. From this table, we can see that our method achieves significant quality improvement over median filtering under all noise levels.

In this supplementary material, we also provide the PSNR/SSIM results for DAVIS testset [14] in Table 3. As Table 3 shows, our method outperforms other methods on high noise levels (σ = 20, 30, 40, and 50) in terms of PSNR and outperforms them on all noise levels in terms of SSIM. With the increasing of noise level , the difference of performance between our method and BasicVSR++ also becomes larger. When σ is 10, 20, 30, 40, and 50, the difference of PSNR between our method and BasicVSR++ is -0.04, 0.03, 0.03, 0.10 and 0.18, respectively. These results demonstrate that the advantage of our neural compression-based feature learning framework is greater when the noise intensity is larger.

Qualitative Comparison. We show another two visual examples in Fig. 5 and Fig. 6. As Fig. 5 shows, other methods cannot restore the bridge deck texture well under this challenging case. The results of FastDVDNet and EMVD suffer from serious distortion and are quite blurry. The results of EDVR, BasciVSR, and BasicVSR++ show better visual quality but the bridge deck textures are still not clear. By contrast, through leveraging better temporal feature alignment and neural compression-based feature learning, our method restores clearer and more accurate bridge deck textures. As Fig. 6 shows, the grass patterns are smoothed in the results of FastDVDNet, EMVD, EDVR, BasicVSR, and BasicVSR++. The results of our method are clearer and the restored grass patterns are more similar to the target.

6. More Comparisons on Video Deraining

We compare our method with prior SOTA video deraining methods, including MS-CSC [10], SE [19], SpacCNN [6], FastDerain [9], J4RNet-P [12], FCRVD [21], RMFD [22], and BasicVSR++ [3]. In the main paper, we present the results on RainSynAll100 [22] and RainSynComplex25 [12]. In this supplementary material, we report the results on NTU-Rain [6] and RainSynLight25 [12]. As Table 4 shows, BasicVSR++ beats RMFD in terms of PSNR and SSIM on both testsets. Compared with BasicVSR++, our method even brings PSNR gain of 0.74 dB and SSIM gain of 0.0035 on NTU-Rain, which shows the benefit of the noise-robust features.

We show visual comparison in Fig. 7. As Fig. 7 shows, BasicVSR++ could not remove the rain streak well and suffers from severe artifacts. RMFD could remove rain streak completely but its results suffer from serious color shading. Instead, our method could produce clearer and visual pleasing results.

7. Visual Comparisons on Video Dehazing

Since the code of CG-IDN [24] is not released, we compare our method with BasicVSR++ [3] and MBSDN [8] in Fig. 8. As Fig. 8 shows, the result of BasicVSR++ contains many artifacts and suffers from poor visual quality. The result of MBSDN is cleaner but its color temperatures are inconsistent with the target. Instead, our method could produce visual pleasing result with accurate color temperatures.

σ	VNLnet [7]	DVDNet [16]	FastDVDNet [17]	EMVD-L [13]	EMVD-S [13]	EDVR [18]	BasicVSR [2]	BasicVSR++ [3]	Ours
10	35.83/0.9473	38.13/0.9679	38.71/0.9702	38.57/0.9695	36.90/0.9512	39.23/0.9732	39.55/0.9758	**39.71**//**0.9761**	**39.67**/**0.9782**
20	34.49/0.9231	35.70/0.9470	35.77/0.9468	35.39/0.9413	33.58/0.9023	36.33/0.9516	36.65/0.9558	**36.75**/**0.9565**	**36.78**/**0.9596**
30	33.42/0.9086	34.08/0.9255	34.04/0.9252	33.89/0.9210	31.94/0.8878	34.62/0.9311	35.07/0.9389	**35.21**/**0.9403**	**35.24**/**0.9435**
40	32.32/0.8974	32.86/0.9040	32.82/0.9047	32.40/0.8941	30.66/0.8508	33.40/0.9113	33.73/0.9207	**33.96**/**0.9229**	**34.06**/**0.9267**
50	31.43/0.8761	31.85/0.8829	31.86/0.8851	31.47/0.8747	29.88/0.8273	32.41/0.8937	32.81/0.9046	**32.93**/**0.9056**	**33.11**/**0.9107**
GFLOPs	-	-	665	1106	5	3089	2947	3402	771

Table 3. PSNR/SSIM comparison with SOTA video denoising methods on DAVIS testset. The best performance is highlighted in **red** (1st best) and **blue** (2nd best). Our method achieves the best SSIM on all noise levels.

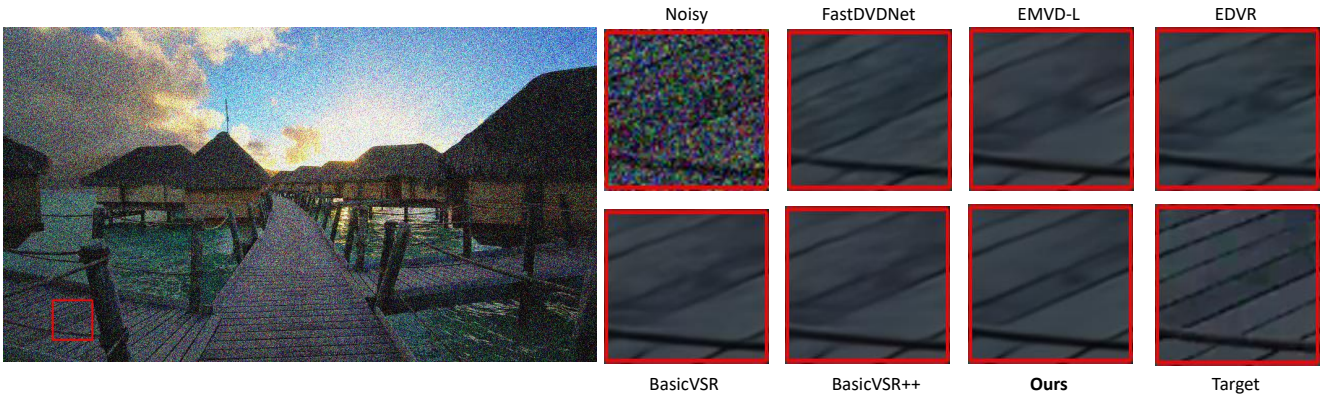

Figure 5. The denoised results of *hypersmooth* from Set8 testset with noise variance 50. Best viewed in color.

References

[1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*, 2015. 1

[2] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4947–4956, 2021. 4, 5

[3] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving Video Super-Resolution with Enhanced Propagation and Alignment. *arXiv preprint arXiv:2104.13371*, 2021. 4, 5, 6

[4] Baymax Chen. EMVD, https://github.com/baymax-chen/emvd, 2021. 4

[5] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018. 1

[6] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6286–6295, 2018. 2, 4, 6

[7] Axel Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. Non-local video denoising by cnn. *arXiv preprint arXiv:1811.12758*, 2018. 4, 5

[8] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2157–2167, 2020. 1, 4

[9] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing*, 28(4):2089–2102, 2018. 4, 6

[10] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653, 2018. 4, 6

[11] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1

[12] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3233–3242, 2018. 2, 4, 6

[13] Matteo Maggioni, Yibin Huang, Cheng Li, Shuai Xiao, Zhongqian Fu, and Fenglong Song. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vi-*

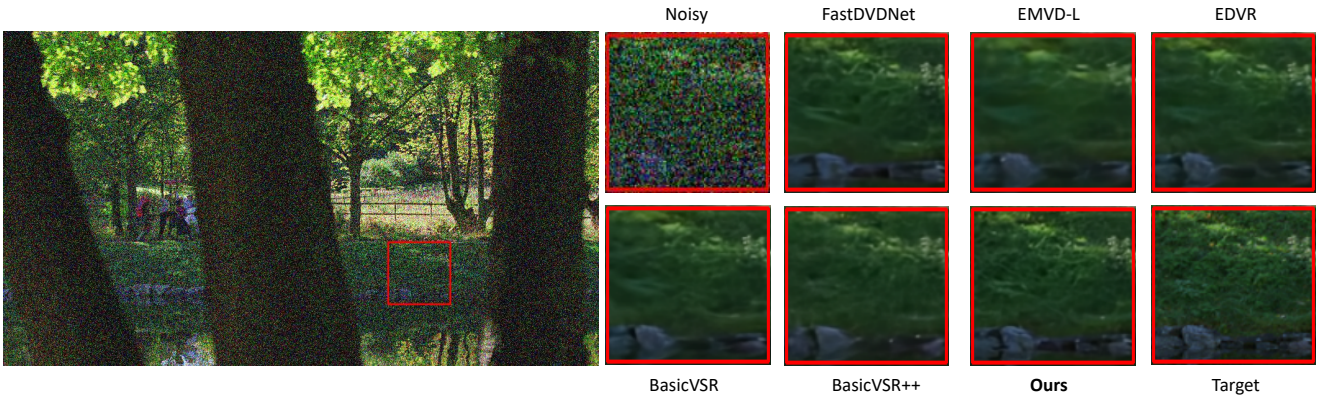

Figure 6. The denoised results of *park_joy* from Set8 test set with noise variance 50. Best viewed in color.

		MS-CSC [10]	SE [19]	SpacCNN [6]	FastDerain [9]	J4RNet-P [12]	FCRVD [21]	RMFD [22]	BasicVSR++ [3]	Ours
NTU-Rain	PSNR	27.31	25.73	33.11	30.32	32.14	36.05	38.92	**39.48**	**40.22**
	SSIM	0.7870	0.7614	0.9474	0.0.9262	0.9480	0.9676	0.9764	**0.9776**	**0.9811**
RainSynLight25	PSNR	25.58	26.56	32.78	29.42	32.96	35.80	36.99	**38.56**	**39.17**
	SSIM	0.8089	0.8006	0.9239	0.8683	0.9434	0.9622	0.9760	**0.9813**	**0.9872**

Table 4. Comparison with SOTA video deraining methods on NTU-Rain [6] and RainSynLight25 [12]. The best performance is highlighted in **red** (1st best) and **blue** (2nd best). We train the BasicVSR++ [3] using the same setting as ours. Other baseline results are provided by RMFD [22] paper.

	σ=10	σ=20	σ=30	σ=40	σ=50
Median filtering (3x3)	27.87	22.45	19.91	18.34	15.95
Median filtering (5x5)	26.63	21.80	19.43	17.92	15.65
Ours	37.12	34.22	32.57	31.39	30.45

Table 5. PSNR comparison with median filtering under AWGN, Set8 testset.

sion and Pattern Recognition (CVPR), pages 3466–3475, June 2021. 4, 5

[14] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 2, 4

[15] 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*, pages 234–241. Springer, 2015. 1

[16] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1805–1809. IEEE, 2019. 5

[17] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1354–1363, 2020. 4, 5

[18] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 4, 5

[19] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2516–2525, 2017. 4, 6

[20] Xide Xia and Brian Kulis. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*, 2017. 1

[21] Wenhan Yang, Jiaying Liu, and Jiashi Feng. Frame-consistent recurrent video deraining with dual-level flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1661–1670, 2019. 4, 6

[22] Wenhan Yang, Robby T Tan, Jiashi Feng, Shiqi Wang, Bin Cheng, and Jiaying Liu. Recurrent multi-frame deraining: Combining physics guidance and adversarial learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 4, 6

[23] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2301–2310, 2020. 2

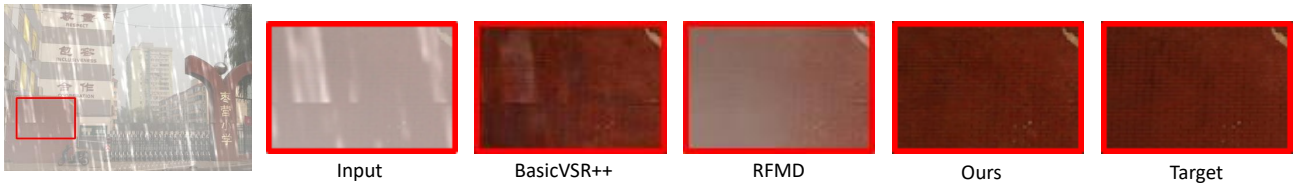

Figure 7. The results of *0995* video from RainSynAll100 testset . Best viewed in color.

Figure 8. The results of *W002* video from REVIDE testset. Best viewed in color.

[24] Xinyi Zhang, Hang Dong, Jinshan Pan, Chao Zhu, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Fei Wang. Learning to restore hazy video: A new real-world dataset and a new method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9239–9248, 2021. 2, 4

[25] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 1