Video Denoising: A Return to Robustness

2025 하계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
신하늘

Outline

- Background
  - Noise in normal light
  - Video Denoising
- Temporal As a Plugin: Unsupervised Video Denoising with Pre-Trained Image Denoisers
  - ECCV 2024
- Classic Video Denoising in a Machine Learning World:Robust, Fast, and Controllable
  - CVPR 2025

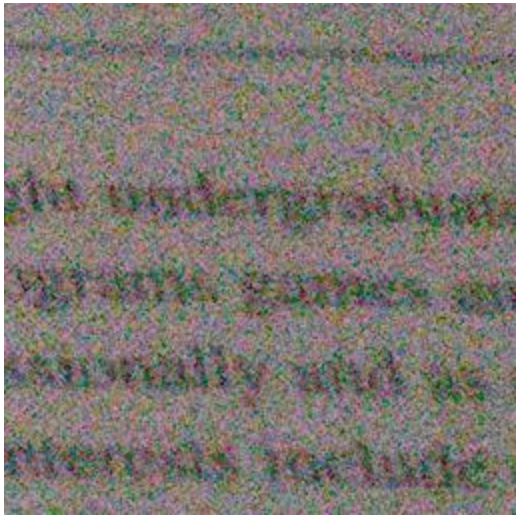
Background

- Noise in normal light
  - $D = \underbrace{K_d K_a (I + N_p)}_{\text{signal-dependent}} + \underbrace{K_d K_a N_1 + K_d N_2 + K_d N_q}_{\text{signal-independent}}\}$
    - $K_a(I + N_p + N_1)$: analog signa이 analog gain을 거치기 전후의 과정에서의 noise
      - I: 실제 장면의 빛 신호 (입사 광자 수)
      - N_p : Photon shot noise
      - N_1 : Analog gain 전에 발생하는 다른 모든 노이즈의 합
      - K_a : Analog gain, ADC에 들어가기 전 신호 증폭
    - $K_d(\ldots + N_2 + N_q)$: Digital 단계에서 신호가 변환되고 증폭되는 과정에서의 noise
      - N_2 : Digital gain 전에 발생하는 다른 모든 노이즈의 합
      - N_q : $Quantization\ noise$
      - K_d : Digital gain

1) NAFNet: Nonlinear Activation Free Network for Image Restoration [ECCV 2022]. Available: https://github.com/megvii-research/NAFNet
2) Classic Video Denoising in a Machine Learning World: Robust, Fast, and Controllable [CVPR 2025]. Available: https://snameo.github.io/projects/levd

Background

- Video Denoising
  - Image Denoising vs Video Denoising
    - Noise in Video = Typical Degradation in Image + 비디오 압축 과정에서 발생하는 노이즈
    - 문제점
      - 인접 프레임 간 Temporal Correlation 때문에 프레임 간 노이즈 연관성 발생
        - ✓H.264 코덱: 인접 프레임 간 정보를 복사하고 공유하는 방식으로 압축 수행
      - 비디오 내에서 발생하는 움직임은 노이즈 패턴에 복잡성 증가

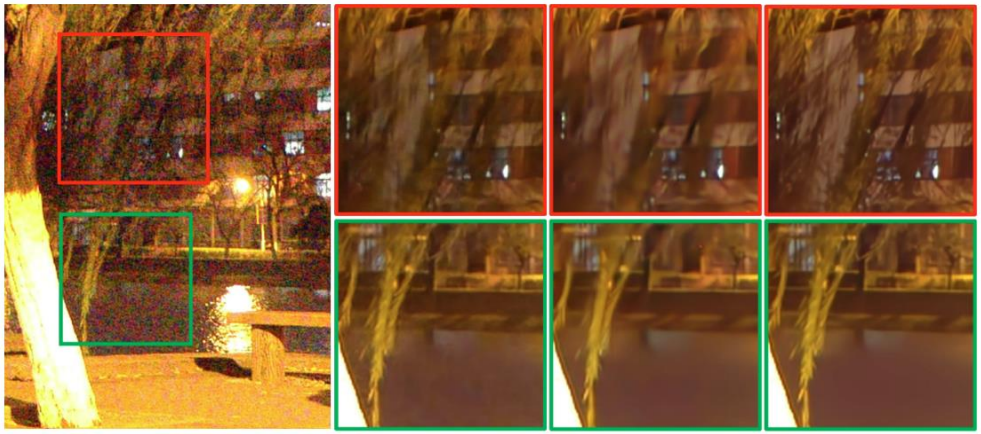


<Image & Video Denoising 적용 예시>

Temporal As a Plugin:
Unsupervised Video Denoising with Pre-Trained Image Denoisers (ECCV 2024)

Introduction

- The Challenges of Existing Video Denoising Methods
  - Supervised Learning:
    - 최근 많은 딥러닝 모델들이 대부분 supervised learning 방식
    - 훈련을 위해서는 많은 수의 paired noisy and clean videos를 필요로 함
      - 움직임이 많은 장면에서의 noisy-clean videos 데이터 쌍을 수집하는 것이 어려워 정지된 순간을 반복 촬영하여 이미지들의 노이즈 프레임을 평균 내고 BM3D 적용하여 GT 프레임 생성[1)]
  - Unsupervised Learning:
    - 위와 같은 이유로 noise videos만으로 학습을 수행하는 unsupervised methods 제안됨
    - 그러나, unsupervised methods는 대부분 artifacts, residual noise 발생



<Fig 1. 기존 video denoising 분야 모델 비교>

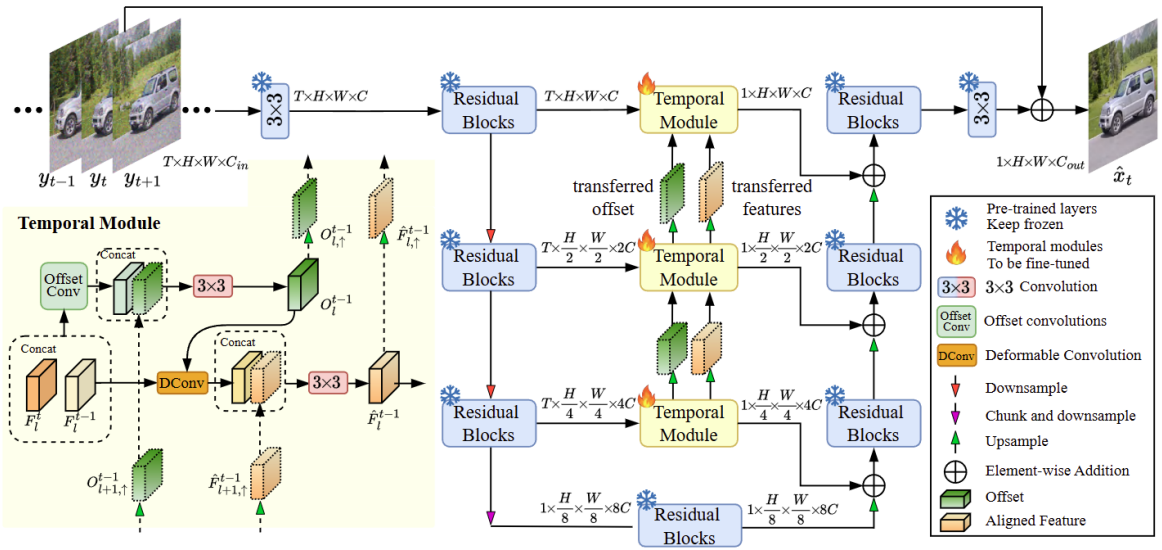
<A paired noisy-clean image>

Related Works

- Unsupervised Video Denoising
  - DNCNN[1)]
    - Model for Image Denoising
    - Convolution, BN(Batch Normalization), ReLU, stacked plain block, global residual connection(잔차 학습)
  - EDVR[2)]
    - Supervised Video Denoising Model
    - PCD(Pyramid, Cascading and Deformable) 정렬 모듈, TSA(Temporal and Spatial Attention) 융합 모듈
    - REDS datasets, Vid4, Vimeo-90K-T datasets 사용
  - Temporal As a Plugin[3)]
    - 동적인 장면 clean-noise 이미지 쌍을 수집하기 어렵기에 Unsupervised 방식의 Video Denoising 프레임워크 제안
    - Pre-trained image denoiser를 기본으로 하여 tunable temporal modules 추가한 형태
    - Temporal module: 여러 노이즈 프레임 간의 공간 복원 시 temporal information을 활용하도록 함

Methodology

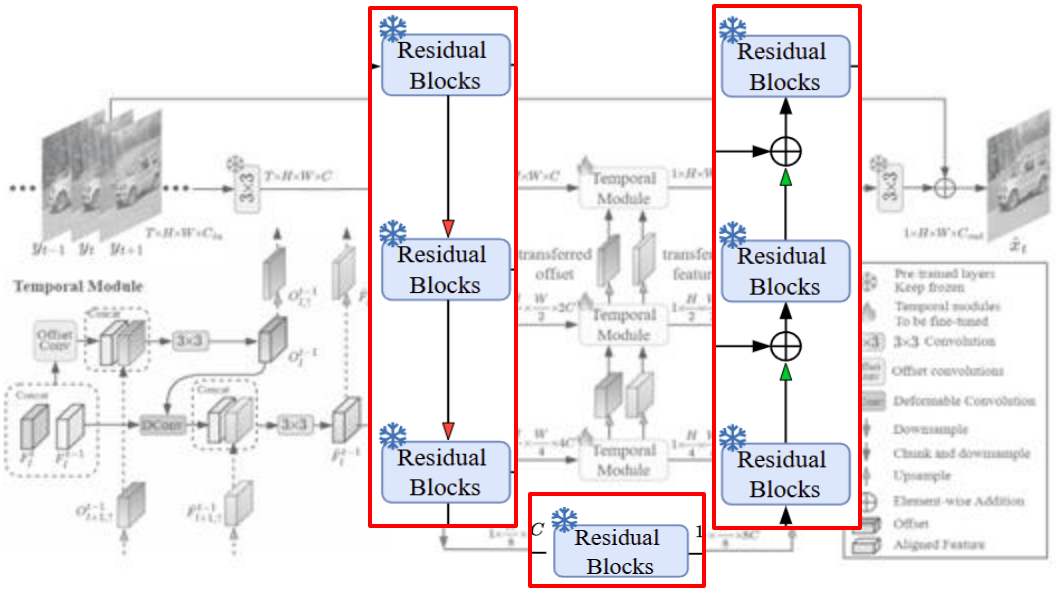
- Overall Architecture of Video Denoiser
  - 사전 학습된 Image denoiser와 Temporal module을 통합하여 구성
  - Input: $Y = \{y_1, \dots y_t, \dots, y_N\}$
    - 비디오 Y에서 T개의 프레임 추출: $Y_t \in R^{T\times H\times W\times C_{in}}$ *실험에서는 T=5
    - 중심 프레임 $y_t \in R^{1\times H\times W\times C_{in}}$을 추출하여 video denoiser의 입력으로 사용
  - Output: $\hat{X} = \{\hat{x}_1, \dots \hat{x}_t, \dots, \hat{x}_N\}$



<Fig 2. 제안한 video denoising overall architecture>

Methodology

- Overall Architecture of Video Denoiser
  - 4-level encoder-decoder 기반 image denoiser
    - 여러 개의 residual block으로 구성
    - Encoder-decoder 사이에 각 skip connection 에 temporal module 삽입
      - spatial information을 보존하면서 temporal information 활용
    - 3 × 3 Convolution 로 특징 추출, residual blocks를 통과하며 feature map 크기 축소(downsample)

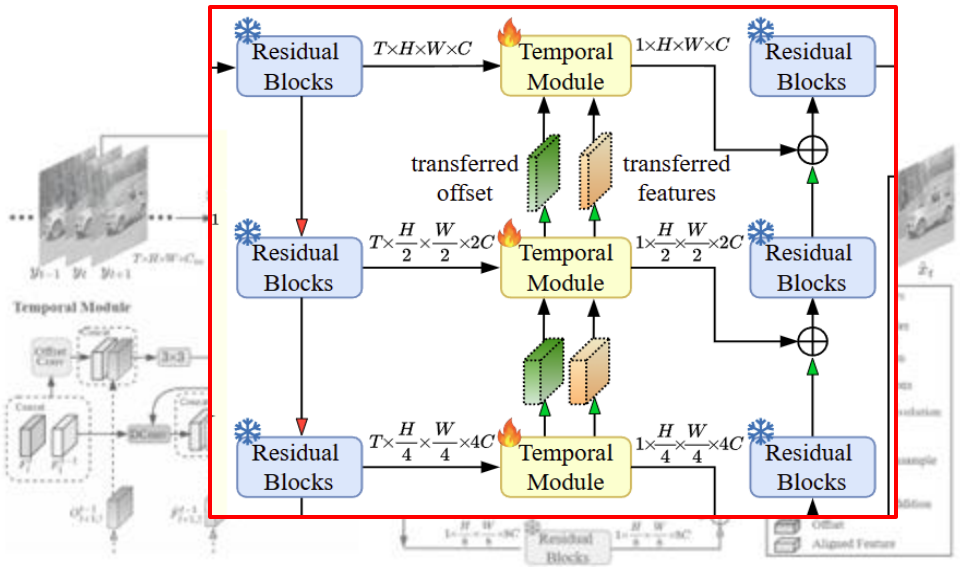


<제안한 video denoising method의 residual blocks 구조>

Methodology

- Temporal Module
  - Level-1부터 Level-3까지 skip connection에 temporal module 연결(Level-4에 X)
  - Residual Blocks를 통해 공간 정보가 들어오면 해당 모듈이 시간 정보를 반영하도록 함
  - Input: $F_l^{t-m} \in R^{1\times H'\times W'\times C'}$
  - Output: $\widehat{F_l} \in R^{1\times H'\times W'\times C'}$

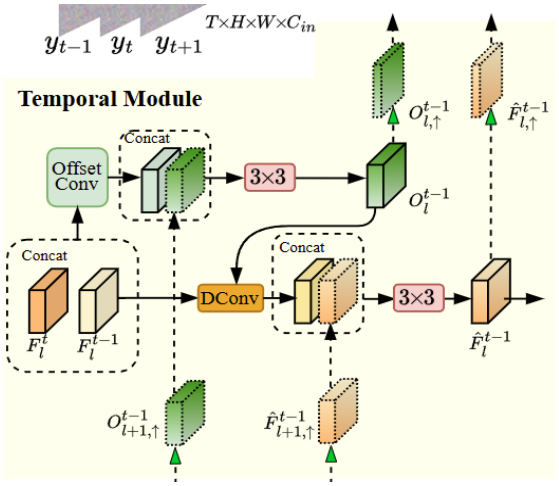
- T : 한 번에 처리하는 프레임 수
- m: -T//2 ~ T//2

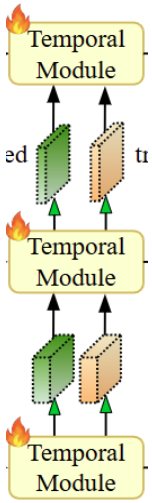


<제안한 video denoising method의 temporal module 구조>

Methodology

- Temporal Module
  - Deformable convolution(DCN)을 적용하여 인접 frame의 중앙 frame의 deep feature를 정렬



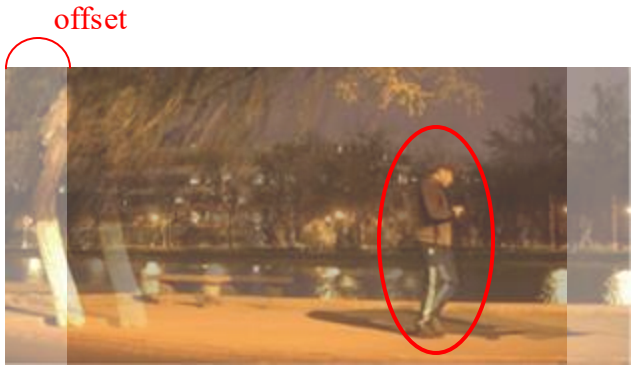


$$\widetilde{O_l^{t-m}} = C(Concat[F_l^t, F_l^{t-m}])$$

$$O_l^{t-m} = Conv\left(Concat\left[\widetilde{O_l^{t-m}}, O_{l+1,\uparrow}^{t-m}\right]\right)$$

Deformable Convolution

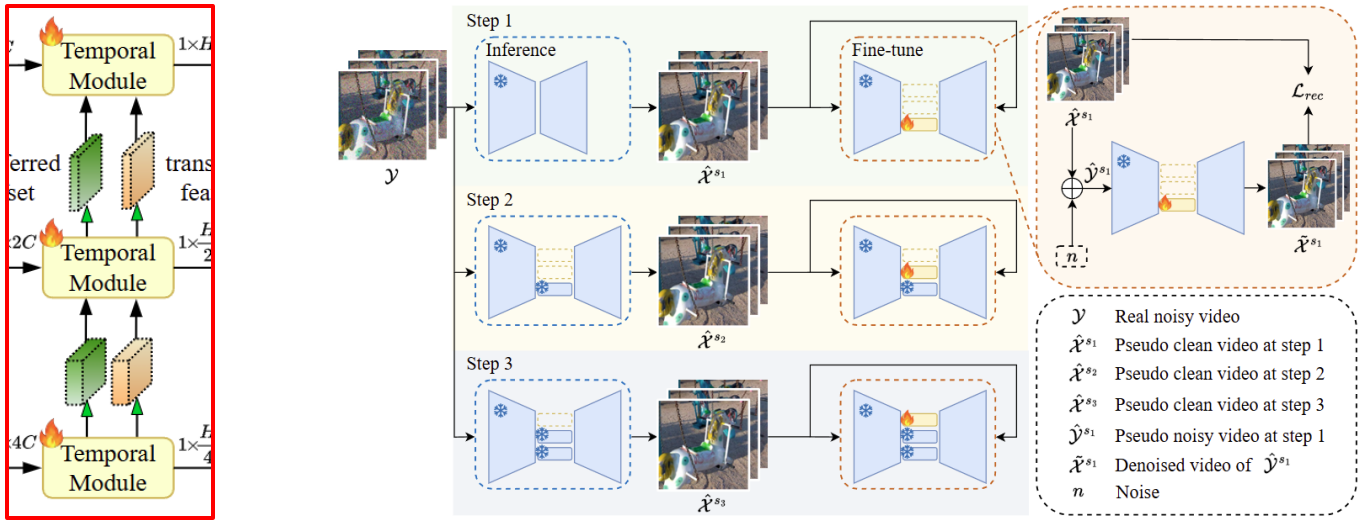
$$\widetilde{F_l^{t-m}} = Conv\left(Concat\left[D(F_l^{t-m}; O_l^{t-m}), \widetilde{F_{l+1,\uparrow}^{t-m}}\right]\right)$$



<temporal module의 offset 학습 예시>

Methodology

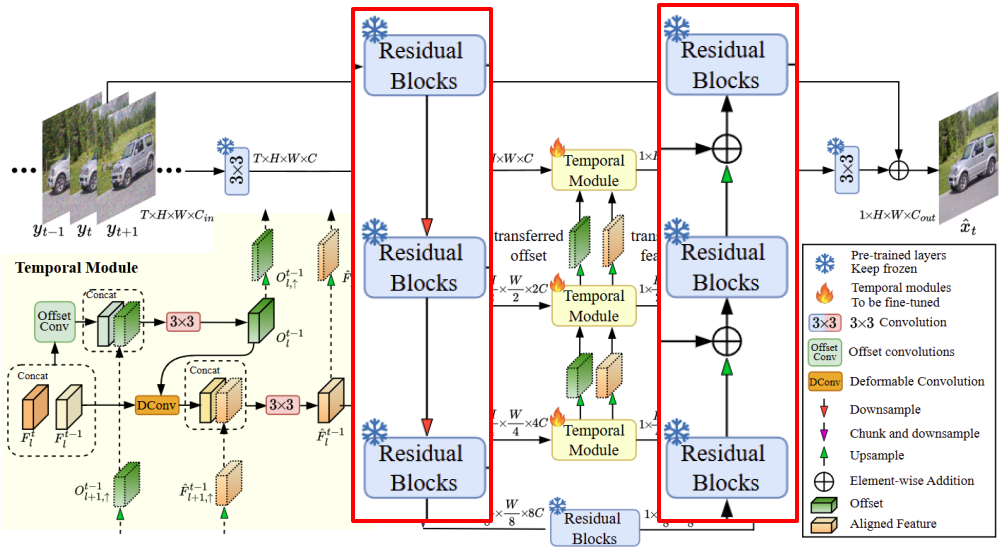
- Unsupervised Progressive Fine-tuning
  - Temporal module을 점진적으로 훈련하기 위한 비지도 fine-tuning 전략
  - Input y로부터 사전 학습된 pre-trained Image Denoiser 를 통해 깨끗한 이미지 $\hat{x}^{s_m}$ 생성
  - $\hat{x}^{s_1}$ 에 인위적으로 노이즈 모델 추가하여 $\hat{y}^{s_m}$ 생성
    - Additive White Gaussian Noise(AWGN)
    - Poisson-Gaussian Noise
  - A paired psuedo noisy-clean video: $\{\hat{y}^{s_m}, \hat{x}^{s_m}\}$
    - $m = \{1,2,3\}$ 순서대로 모듈이 상위 레벨로 올라가며 fine – tuning



<temporal module fine-tuning 과정>

Experiment

- Implementation Detail
  - Video Denoiser: NAFNet(Encoder-Decoder)을 baseline image denoiser로 사용
    - sRGB, raw 이미지용 denoiser 개별 학습 데이터셋
      - sRGB: BSD500, DIV2K, Flickr2K 및 WED dataset
        - ✓Additive White Gaussian Noise(AWGN) 노이즈 합성
      - Raw: SID, SIDD, CRVD dataset
        - ✓Poisson-Gaussian 노이즈 합성



<NAFNet 기반 image denoiser 구조>

Experiment

- Experimental on synthetic Gaussian video denoising
  - Supervised method 의 FloRNN 다음으로 좋은 성능을 나타냄
  - 모델 이름이 –T 로 끝나는 경우 test dataset으로 직접 fine-tuning함
  - DAVIS dataset에 비해 Set8 dataset이 부드러운 움직임을 가짐
    - TAP-T와 기타 모델의 데이터셋 별 결과가 DAVIS 에 비해 Set8이 더 차이가 큼
    - 부드러운 움직임 = Temporal information을 더 효과적으로 인식 가능

		Method	$\sigma = 10$		$\sigma = 20$		$\sigma = 30$		$\sigma = 40$		$\sigma = 50$		Average	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DAVIS	Traditional	VBM4D [29]	37.58	-	33.88	-	31.65	-	30.05	-	28.80	-	32.39	-
	Supervised	NAFNet [8]	38.79	0.965	35.37	0.933	33.47	0.904	32.17	0.879	31.18	0.858	34.20	0.908
		FastDVDNet [42]	38.71	0.962	35.77	0.941	34.04	0.917	32.82	0.895	31.86	0.875	34.64	0.919
		PaCNet [43]	39.97	0.971	37.10	0.947	35.07	0.921	33.57	0.897	32.39	0.874	35.62	0.922
		FloRNN [21]	40.16	0.976	37.52	0.956	35.89	0.944	34.66	0.929	33.67	0.913	36.38	0.944
	Unsupervised	MF2F-T [11]	38.04	0.957	35.61	0.936	33.65	0.907	31.50	0.852	29.39	0.784	33.64	0.887
		RFR-T [18]	39.31	-	36.15	-	34.28	-	32.92	-	31.86	-	34.90	-
		UDVD [38]	39.17	0.970	35.94	0.943	34.09	0.918	32.79	0.895	31.80	0.874	34.76	0.920
		RDRF [47]	39.54	0.972	36.40	0.947	34.55	0.925	33.23	0.903	32.20	0.883	35.18	0.926
		ER2R-T [59]	39.52	-	36.49	-	34.60	-	33.29	-	32.25	-	35.23	-
		TAP	39.69	0.972	36.62	0.948	34.71	0.925	33.37	0.903	32.36	0.884	35.35	0.926
		TAP-T	**39.80**	**0.973**	**36.74**	**0.950**	**34.84**	**0.926**	**33.49**	**0.905**	**32.49**	**0.886**	**35.48**	**0.928**
Set8	Traditional	VBM4D [29]	36.05	-	32.19	-	30.00	-	28.48	-	27.33	-	30.81	-
	Supervised	NAFNet [8]	36.52	0.943	33.55	0.802	31.81	0.869	30.59	0.842	29.65	0.818	32.43	0.875
		FastDVDNet [42]	36.44	0.954	33.43	0.920	31.68	0.889	30.46	0.861	29.53	0.835	32.31	0.892
		PaCNet [43]	37.06	0.960	33.94	0.925	32.05	0.892	30.70	0.862	29.66	0.835	32.68	0.895
		FloRNN [21]	37.57	0.964	34.67	0.938	32.97	0.914	31.75	0.891	30.80	0.870	33.55	0.915
	Unsupervised	MF2F-T [11]	36.01	0.938	33.79	0.912	32.20	0.883	30.64	0.841	28.90	0.778	32.31	0.870
		RFR-T [18]	36.77	-	33.64	-	31.82	-	30.52	-	29.50	-	32.45	-
		UDVD [38]	36.36	0.951	33.53	0.917	31.88	0.887	30.72	0.859	29.81	0.835	32.46	0.890
		RDRF [47]	36.67	0.955	34.00	0.925	32.39	0.898	31.23	0.873	30.31	0.850	32.92	0.900
		ER2R-T [59]	37.55	-	34.34	-	32.45	-	31.09	-	30.05	-	33.10	-
		TAP	37.76	0.956	34.51	0.925	32.76	0.899	31.21	0.873	30.27	0.851	33.30	0.901
		TAP-T	**38.02**	**0.958**	**35.07**	**0.927**	**33.42**	**0.900**	**32.10**	**0.875**	**31.16**	**0.852**	**33.95**	**0.902**

<Fig 1. Quantitative evaluation(using PSNR and SSIM) of video denoising results on the DAVIS and Set8 datasets>

Experiment

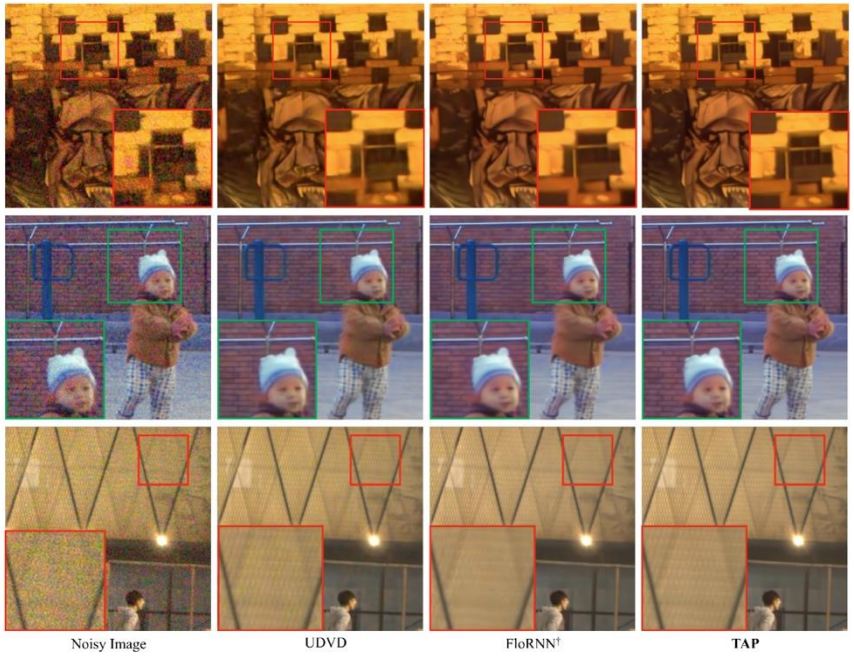
- Experiments on real raw video denoising
  - Indoor용 CRVD dataset 사용(raw image)
    - Dataset의 크기가 작고, 수동으로 조작한 작위적인 움직임을 갖고 있음
    - Outdoor용 dataset보다는 테스트 시 중요도 떨어짐

	Method \ ISO	1600	3200	6400	12800	25600	Average
Supervised	NAFNet [8]	48.02	46.05	44.04	41.44	41.49	44.21
	RViDeNet [53]	47.74	45.91	43.85	41.20	41.17	43.97
	FloRNN [21]	48.81	47.05	45.09	42.63	42.19	45.15
Unsupervised	UDVD [38]	48.02	46.44	44.74	42.21	42.13	44.71
	RDRF [47]	48.38	46.86	**45.24**	**42.72**	42.25	45.09
	Ours	**48.85**	**47.03**	45.11	42.44	**42.33**	**45.15**

<Tab 2. Quantitative evaluation (PSNR in dB) of raw video denoising results on the CRVD indoor test set>

Experiment

- Visual Examples of real raw video denoising
  - Outdoor용 CRVD dataset 사용(raw image)
    - Supervised method인 FloRNN은 과도하게 부드러운 결과 생성
    - Unsupervised method인 UDVD는 노이즈 제거 능력이 떨어짐
    - 논문에서 제안한 TAP 모델은 FloRNN 만큼의 결과 도출



<Fig 5. Visual examples of real raw video denoising on CRVD outdoor set>

Ablation Study

- Ablation on different Fine-tuning strategies
  - Experiments are conducted on the DAVIS dataset with noise level σ = 30.
    - Original: fine-tuning temporal modules progressively
    - Case 1: fine-tuning all parameters progressively
    - Case 2: fine-tuning three temporal modules jointly
    - Case 3: repeat the fine-tuning process twice

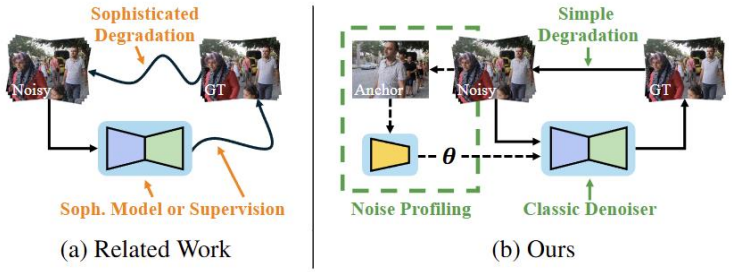
	Step 0	Step 1	Step 2	Step 3
Original	33.47 dB	34.47 dB	34.77 dB	34.84 dB
Case 1	33.47 dB	34.02 dB	34.14 dB	34.20 dB
Case 2	33.47 dB	34.49 dB	34.62 dB	34.55 dB
Case 3	-	34.54 dB	34.81 dB	34.86 dB

<Tab 3. Ablation on different fine-tuning strategies>

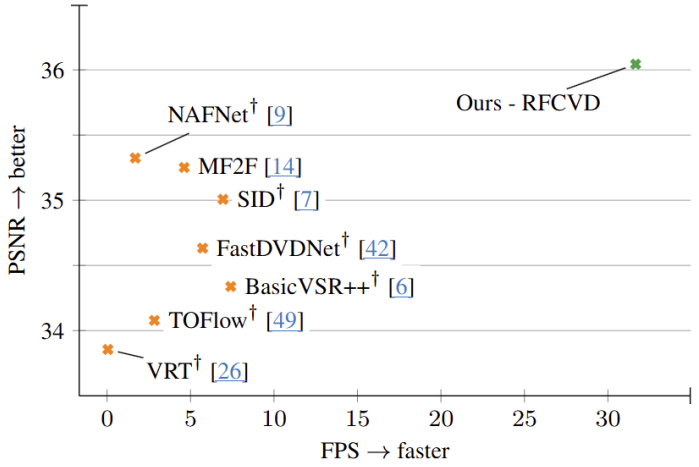
Classic Video Denoising in a Machine Learning World: Robust, Fast, and Controllable (CVPR 2025)

Introduction

- Deep-learning based Video Denoising model
  - 노이즈 제거 품질 향상됐으나 robust하지 못함
    - Compression artifacts처럼 temporally correlated한 noise는 noise pattern이 다양해 학습이 어려움
    - Training data에 크게 의존하는 경향을 보임
- Classic Video Denoising machine learning model
  - Real-world의 복잡한 video noise에도 robust하게 작동하고 속도가 빠름
    - 본 논문에서는 differentiable denoising pipeline based on traditional methods을 제시
      - Pipeline에 필요한 최적의 parameter를 neural network로 예측



<Fig 2. High-level comparison of how related work approaches>



<Fig 3. Video denoising on the CRVD (sRGB) benchmark>

Related Works

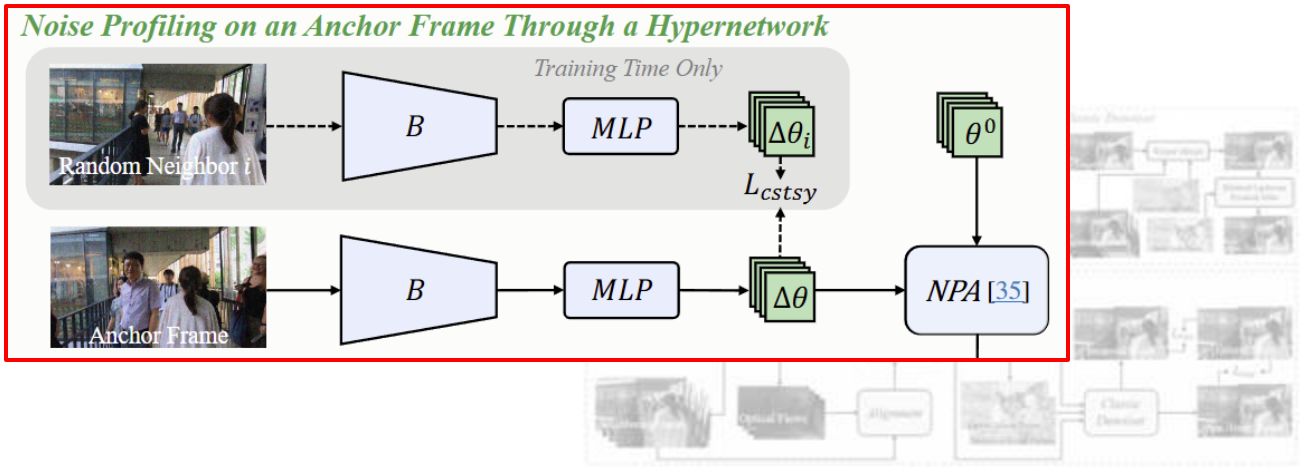
- Video Denoising Pipeline
  - Classical Video Denoising
    - Block matching 기반의 다양한 denoising 모델 등장
    - 특히 bilateral filtering, image pyramids의 개념을 통합하기도 함
  - Machine Learning 기반 Video Denoising
    - Explicit Matching / Implicit Correspondence
    - 학습한 적 없는 노이즈 패턴은 제거 불가능
    - Noisy-clean video 데이터를 수집하기 어려워 self-supervised learning 도 제안됨
  - Noise Simulation
    - GT(Ground Truth)에 인위적인 노이즈를 추가하여 noisy-clean video 데이터를 만듦
      - 이때 노이즈 제거 능력은 향상되지만, 일반화 성능은 떨어짐
    - 이때 training data와 실제 비디오의 다양한 noise patterns 간의 불일치 발생

Method

Consistency loss

$$\mathcal{L}_{cstsy} = \|\Delta\theta - \Delta\theta_i\|_2$$

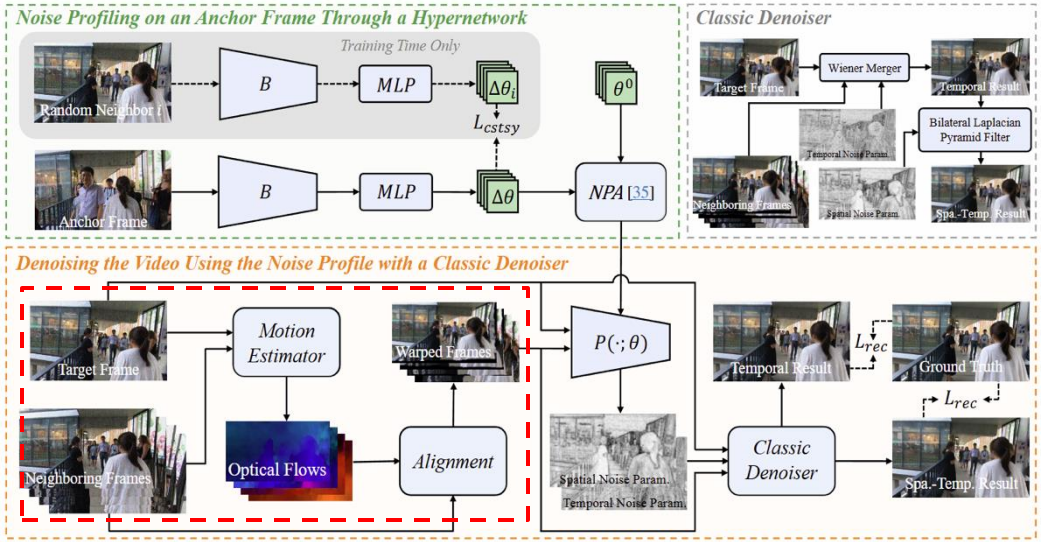
- Noise Profiling
  - Video의 noise profile은 시간이 지나도 변하지 않다는 점을 이용해 noise를 제거하기 전에 noise profiling 하는 단계를 추가
    - 하나의 anchor frame을 선택하여 noise profile(θ) 추정
      - $\theta = \theta^0 + \Delta\theta \leftarrow$ hypernetwork로 $\Delta\theta$를 예측(NPA)
        - ✓NPA: 안정적인 예측을 위해 메인 네트워크의 가중치(θ)를 직접 예측하는 대신, $\Delta\theta$를 예측
      - P(·; θ): denoising parameter를 예측하는 neural network, 추후 프레임에 적합한 noise map 예측
      - Anchor frame은 video의 첫 frame으로 선택하고 consistency loss 사용하여 일관성 유지
    - B: 특징 추출용 neural network, Backbone은 ConvNext 사용



<Fig 4. Noise Profiling on an Anchor Frame Through a Hypernetwork>

Method

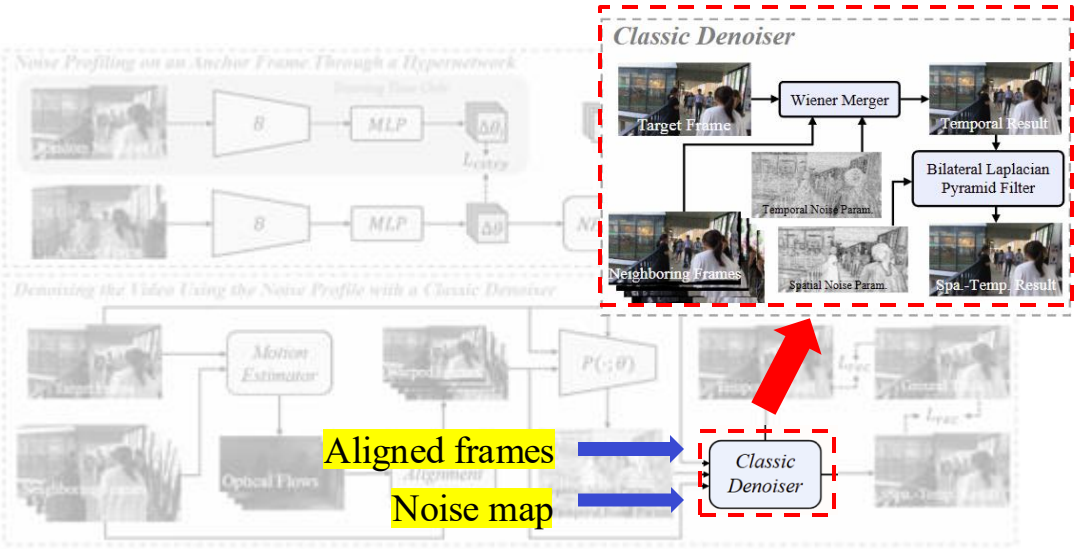
- Denoising the Video
  - Optical flow를 추출하여 frame을 align한 후 temporal noise 제거
    - Target frame이 있으면, 앞뒤로 2개씩 인접 frame 추출
    - SpyNet 모델로 Optical flow 추출, 해상도 ¼로 줄여서 노이즈에 대한 robustness, 계산 효율성 향상
    - 추출한 Optical flow로 target – 인접 frame 간 alignment
      - 움직임이 있는 동일한 피사체의 픽셀을 겹치도록 정렬
    - Align된 frames은 추후 Wiener filter의 input으로 제공됨
      - Random하고 temporal한 특성을 가진 noise를 align한 frames을 평균 내 노이즈 제거



<Fig 4. Class Denoiser on an Anchor Frame Through a Hypernetwork>

Method

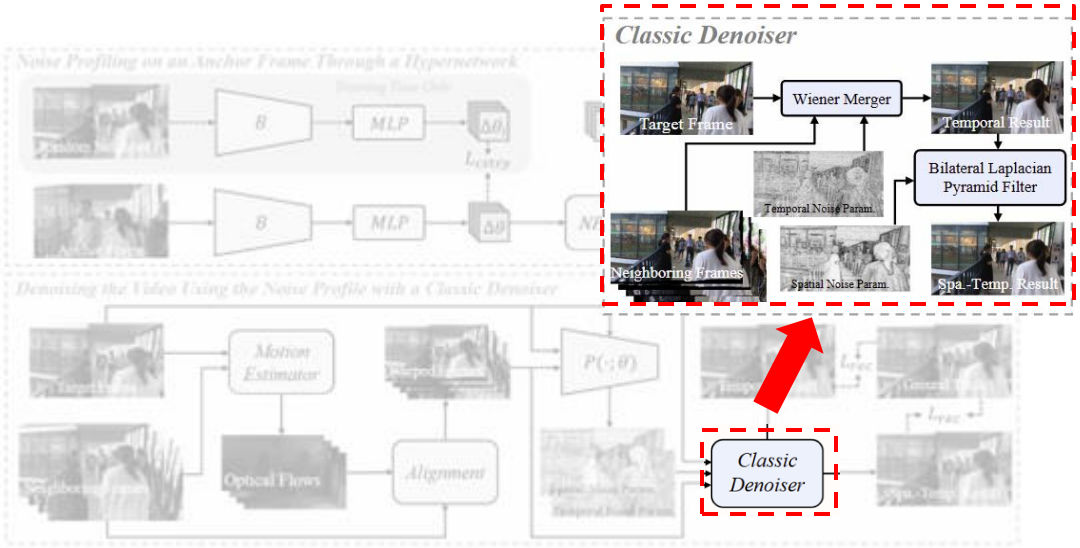
- Denoising the Video
  - 추정된 noise profile(θ)을 사용하여 정렬된 frames로부터 하나의 clean frame 추출
    - P(·; θ)로부터 noise의 분산 σ^2을 나타내는 noise map을 Wiener Merger에 전달
    - Wiener Merger
      - 일반적으로 Wiener filter는 모든 픽셀에 대해 동일한 노이즈 레벨이라 가정하고 노이즈 제거
      - 5개의 frame마다 동일한 픽셀 위치에서 noise map(σ^2)과 신호의 분산(σ^2)을 비교
        ✓노이즈 분산이 크면 노이즈가 많으므로 강한 필터링 적용
      - 이와 같은 방법으로 temporal noise 제거하면서 aligned frames를 하나로 merge



<Fig 4. Class Denoiser on an Anchor Frame Through a Hypernetwork>

Method

- Denoising the Video
  - Temporal noise가 제거된 frame에 Bilateral Laplacian Pyramid Filter 적용
    - Bilateral Laplacian Pyramid filter
      - Spatial noise 제거하기 위해 Bilateral filter + Laplacian pyramid(3 levels) 결합한 filter
      - $P(\cdot;\theta)$이 만든 noise map을 bilateral filter의 σ_S(Spatial Sigma), σ_r(Range Sigma)로 사용
        - ✓σ_S(Spatial Sigma): 가까울수록 높은 가중치
        - ✓σ_r(Range Sigma) : 밝기, 색상이 비슷할수록 높은 가중치
    - Edge만 남기고 다른 부분은 blurring하여 spatial noise 제거



<Fig 4. Class Denoiser on an Anchor Frame Through a Hypernetwork>

Experiments

- Results on the CRVD benchmark
  - 실제 카메라로 촬영한 noise가 있는 sRGB dataset으로 정량적 평가
    - 복잡하고 예측 불가능한 real-world의 noise에 대해 얼마나 잘 작동하는지 확인
    - ISO를 다섯 가지로 분류하여 실험 진행
  - RFCVD가 31.66FPS까지 측정됨 (RTX 3090 GPU)
  - Classic denoising 기술을 사용하여 generalizability를 향상시켜 real-world의 noise 패턴을 파악할 수 있다고 주장

- †: 재학습 모델

	ISO 1600		ISO 3200		ISO 6400		ISO 12800		ISO 25600		Overall		Speed	
	PSNR (higher PSNR is better)	rank	PSNR (higher PSNR is better)	rank	PSNR (higher PSNR is better)	rank	PSNR (higher PSNR is better)	rank	PSNR (higher PSNR is better)	rank	PSNR (higher PSNR is better)	rank	FPS (higher FPS is better)	rank
SID† [7]	38.85	7th of 10	37.68	7th of 10	35.82	4th of 10	33.51	4th of 10	29.18	3rd of 10	35.01	4th of 10	6.95	3rd of 10
NAFNet† [9]	39.48	3rd of 10	38.12	5th of 10	35.94	3rd of 10	33.53	3rd of 10	29.55	2nd of 10	35.32	2nd of 10	1.69	7th of 10
FastDVDNet† [42]	39.16	5th of 10	37.92	6th of 10	35.60	5th of 10	32.56	5th of 10	27.93	6th of 10	34.63	5th of 10	5.72	4th of 10
TOFlow† [49]	38.25	8th of 10	36.97	8th of 10	34.90	7th of 10	32.21	6th of 10	28.07	5th of 10	34.08	7th of 10	2.84	6th of 10
BasicVSR++† [6]	39.40	4th of 10	38.24	2nd of 10	35.55	6th of 10	31.72	7th of 10	26.78	9th of 10	34.34	6th of 10	7.41	2nd of 10
VRT† [26]	39.55	2nd of 10	38.12	4th of 10	34.82	8th of 10	30.77	8th of 10	26.01	10th of 10	33.86	8th of 10	0.05	10th of 10
Real-ESRGAN [46]	29.98	10th of 10	28.27	10th of 10	28.04	10th of 10	27.52	10th of 10	27.63	8th of 10	28.29	10th of 10	0.24	8th of 10
UDVD [40]	31.15	9th of 10	30.72	9th of 10	30.23	9th of 10	29.10	9th of 10	27.63	7th of 10	29.77	9th of 10	0.16	9th of 10
MF2F [14]	39.09	6th of 10	38.20	3rd of 10	36.36	1st of 10	33.57	2nd of 10	29.04	4th of 10	35.25	3rd of 10	4.62	5th of 10
Ours - RFCVD	40.35	1st of 10	38.60	1st of 10	36.28	2nd of 10	33.86	1st of 10	31.12	1st of 10	36.04	1st of 10	31.66	1st of 10

<Tab 2. Video denoising results on the CRVD(sRGB) benchmark>

Experiments

- Test for Robustness and Generalizability
  - 기존 deep-learning 기반 denoising 모델의 'training data에 없는 noise pattern' = 'Domain GAP'에 얼마나 잘 대처하는지 검증
  - Training data
    - AWGN과 H.264 compression noise를 입힌 image
  - Test data
    - Test1: film grain noise + AV1 compression
    - Test2: spatially correlated noise(gaussian) + H.265 compression

	film grain noise w/ AV1 compression						spatially correlated noise w/ H.265 compression						Speed	
	PSNR (higher PSNR is better)	rank	SSIM (higher SSIM is better)	rank	LPIPS (lower LPIPS is better)	rank	PSNR (higher PSNR is better)	rank	SSIM (higher SSIM is better)	rank	LPIPS (lower LPIPS is better)	rank	FPS (higher FPS is better)	rank
SID† [7]	27.05	5th of 7	0.664	5th of 7	0.293	4th of 7	28.44	3rd of 7	0.771	4th of 7	0.282	5th of 7	15.00	3rd of 7
NAFNet† [9]	27.16	4th of 7	0.672	4th of 7	0.294	5th of 7	28.40	4th of 7	0.764	6th of 7	0.257	3rd of 7	3.812	6th of 7
FastDVDNet† [42]	27.39	3rd of 7	0.686	3rd of 7	0.272	3rd of 7	27.92	6th of 7	0.769	5th of 7	0.296	6th of 7	12.09	4th of 7
TOFlow† [49]	28.15	2nd of 7	0.750	2nd of 7	0.220	1st of 7	28.39	5th of 7	0.788	3rd of 7	0.258	4th of 7	5.665	5th of 7
BasicVSR++† [6]	26.90	6th of 7	0.651	6th of 7	0.313	6th of 7	27.48	7th of 7	0.728	7th of 7	0.358	7th of 7	15.32	2nd of 7
VRT† [26]	26.55	7th of 7	0.629	7th of 7	0.331	7th of 7	28.78	2nd of 7	0.803	2nd of 7	0.206	1st of 7	0.105	7th of 7
Ours - RFCVD	28.59	1st of 7	0.774	1st of 7	0.247	2nd of 7	28.93	1st of 7	0.808	1st of 7	0.239	2nd of 7	69.73	1st of 7

<Tab 3. Video denoising results on REDS dataset w/AWGN & H.264 benchmark>

감사합니다