

# A Bit is All You Need! Efficient Video Capture via Single Bit Imaging

## Supplementary Material

### 1. Additional Results

#### 1.1. Effect of Noise

**Training with Noisy Data** We experimented with introducing noise while training an EfficientSCI++. For this experiment, we start with the final model checkpoint trained on clean GoPro data and finetune the model to recover clean ground truth from noisy 1-bit measurements obtained by empirical thresholds on the ground truth corrupted by additive Gaussian noise. We fine-tune the model on varying levels of additive Gaussian noise where the standard deviation  $\sigma \sim \mathcal{U}(0, 0.1)$ . This results in an improved performance across noise levels as seen in Tab. 1.

A sample qualitative result is provided in Fig. 1. Interestingly, the model fine-tuned on noisy data produces quantitatively better reconstructions on the noisy data than the original model on clean inputs. A reason for the better performance could be resulting from the better initialization fine-tuning. Additionally, noise in input frames ensures spatially more frequent threshold crossings such that the resulting 1-bit frames have more information about the scene than the 1-bit frames obtained from clean data, see Fig. 2. Therefore, reconstruction from noisy 1-bit frames becomes easier compared to recovery from clean 1-bit frames. We also tested the noise fine-tuned model on hardware thresholded frames, which yielded a PSNR/SSIM of 30.77/0.9074. These results could further improve by training on hardware generated data.

$\sigma$	Network	PSNR/SSIM <sub>overlap</sub>	PSNR/SSIM <sub>no-overlap</sub>
0	EfficientSCI++	31.27/0.9106	30.75/0.9031
0.01		31.96/0.9195	31.43/0.9125
0.02		32.44/0.9238	31.93/0.9175
0.03		32.77/0.9248	32.29/0.9191
0.05		33.02/0.9222	32.60/0.9173
0.08		32.77/0.9135	32.42/0.9090
0.1		32.43/0.9063	32.11/0.9017
real ADC		30.77/0.9074	30.33/0.8999

Table 1. Reconstruction performance of Efficientnet++ finetuned on noisy input

#### 1.2. Further Qualitative Results

Qualitative results comparing frames reconstructed by Aswin [4] using different threshold methods for three sample video from GoPro test set is provided in Fig. 3 along with corresponding error maps. High quality frames can be reconstructed from even 1-bit measurements for all the methods. The errors in very bright regions are higher with

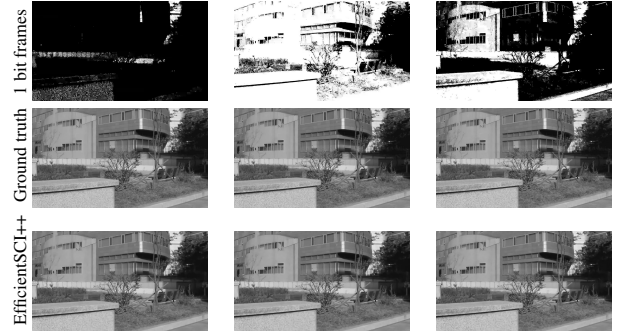


Figure 1. Reconstruction from noisy 1-bit measurements.

Model	FLOPs	#Params	Runtime (ms)
Aswin	2.36 TF	8.91M	237.80±0.99
EfficientSCI++	1.33 TF	8.08M	59.70±0.12
Res2former	1.70 TF	17.70M	104.15±0.39

Table 2. Computational complexity in terms of number of flops, and run-times are shown.

thresholds obtained using beta and empirical distributions, as these thresholds are more concentrated in the regions with higher probability densities.

#### 1.3. Computational Complexity

We measured computational complexity and runtime for recovering a video patch of size  $20 \times 128 \times 128$  on a AMD Ryzen 9 5900 X machine with RTX 3090 GPU, the corresponding values are reported in Tab. 2.

Aswin [4] is the most computationally intensive model requiring 2.36 TFLOPs to reconstruct a video patch of size  $20 \times 128 \times 128$ . EfficientSCI++ [1] achieves a slightly lower performance using significantly lower computational complexity and run-times.

#### 1.4. Description of Videos

We additionally provide videos demonstrating our 1-bit video acquisition and recovery in the attached folder ‘videos’. The videos in the folder ‘Aswin\_res\_videos’ contain videos providing input 1-bit measurements using empirical thresholds, ground truth, and reconstructions using the Aswin model for 11 test videos in GoPro testset. The videos demonstrating the ood robustness of our 1-bit acquisition scheme together with Aswin model for reconstruction are provided in the folder ‘ood\_videos’. The model is trained on GoPro using uniformly spaced non-smooth thresholds and tested on XVFI [6] and QUIVER [2] datasets



Figure 2. 1-bit measurements on GoPro test set for clean ground truth frames and frames corrupted by 5% Gaussian noise.

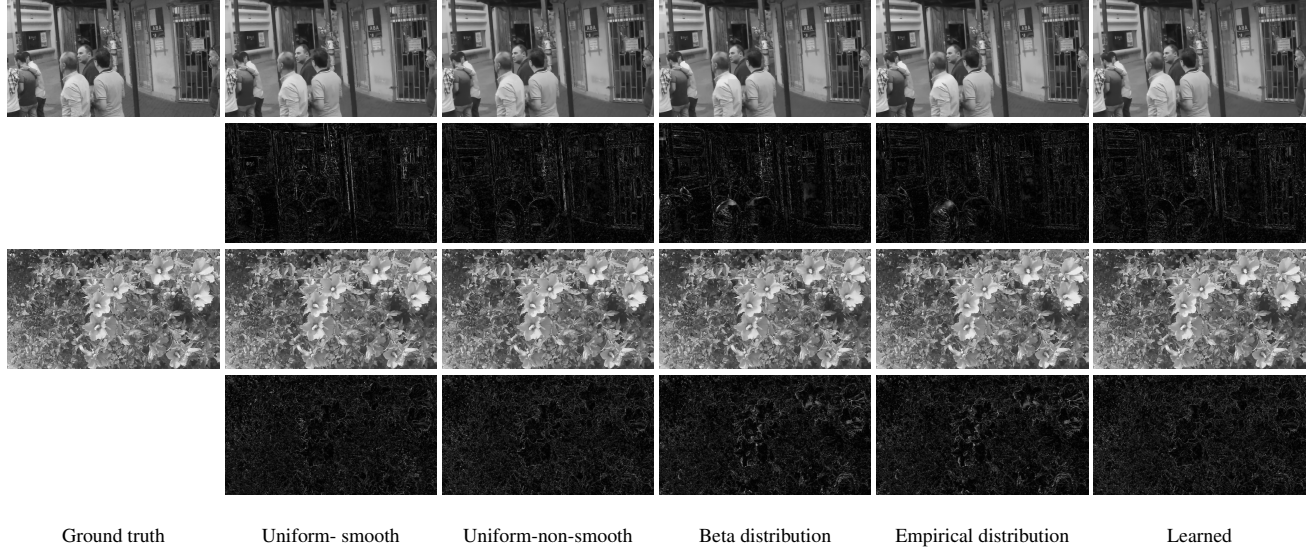


Figure 3. Sample video recovery from 1-bit measurements on GoPro test set. Ground truth frame and reconstructions using different threshold strategies are visualized in the top row, error maps corresponding to reconstructions with error amplified by a factor of 5 are visualized in the bottom row.

in addition to high speed videos from youtube. The sub-folders ‘xvfi’, ‘quiver’ and ‘youtube’ in ‘ood\_videos’ contain sample video reconstructions from the corresponding datasets. Finally, the folder ‘videos\_noise\_train\_test’ shows reconstructions of clean video from noisy measurements (5% Gaussian noise) using the model fine-tuned on noisy data.

## 2. Training settings

We train the different models Aswin [4], EfficientSCI++ [1], Shiftnet [3] Res2former [7] on GoPro dataset [5] using PyTorch version 2.0.1 in half precision (bfloat16) due to memory consideration. We use the model architectures from their publicly available implementations<sup>1234</sup>. All the models except Shiftnet [3] are trained on a machine with two NVIDIA GeForce RTX 3090 GPUs and AMD Ryzen 9 3950X 16-Core Processor. Shiftnet [3] was trained on a machine with a single NVIDIA GeForce RTX 4090 GPU and Intel Core i9-10900K CPU with a clock speed of 3.70GHz. Training batch sizes for each network are selected to opti-

mally utilize available GPU memory. We explore the effect of different threshold selection methods, length of thresholds and scaling of inputs using only Aswin [4] model, for which we use a fixed seed of 8 for fair comparison between the different settings. All the models are trained on GoPro dataset [5] containing 33 videos of which we use 22 videos for training set and 11 videos as test set.

**Aswin** [4] model was trained for 28K epochs with a batch size of 2 (11 iterations per epoch) on randomly cropped gray scale patches of resolution  $20 \times 160 \times 160$  and a starting learning rate of  $5e-4$  using Adam optimizer. Learning rate is decreased to  $4.5e-4$ ,  $1e-4$  and  $3e-5$  at 16K, 22K and 25K epochs.

**EfficientSCI++** [1] model was trained for 28K epochs with a batch size of 6 on randomly cropped gray scale patches of resolution  $20 \times 160 \times 160$  and a starting learning rate of  $3e-4$ , using Adam optimizer. Learning rate is decreased to  $2e-4$ ,  $1e-4$ ,  $5e-5$  and  $3e-5$  at 5.4K, 10.8K, 16.2K and 22K epochs.

**Shiftnet** [3] We change the number of channels from 3 to 1 to process and recover single gray channel videos. To exploit the information from all the thresholds similar to the other benchmarked methods, we utilize the context of 10 future frames and 10 past frames in grouped temporal shifts instead of the default setting of 1 past frame and 1 future

<sup>1</sup><https://github.com/LLindn/ASwin-Video-Denoising>

<sup>2</sup><https://github.com/mcao92/EfficientSCI-plus-plus>

<sup>3</sup><https://github.com/dasongli1/Shift-Net/>

<sup>4</sup><https://github.com/pwangcs/DeepOpticsSCI>

frame used in [3], and process 30 frames together to predict the 10 central frames. As the network architecture is designed to process a batch size of 1, we use batch size of 1 of randomly cropped gray scale patches of resolution  $30 \times 160 \times 160$  and train Shiftnet for 28K epochs (22 iterations per epoch) with a starting learning rate of  $3e - 4$ , using Adam optimizer. Learning rate is decreased to  $2e - 4$ ,  $1e - 4$ ,  $5e - 5$  and  $3e - 5$  at 5.4K, 10.8K, 16.2K and 22K epochs.

**Res2former** [7] model was trained for 28K epochs with a starting learning rate of  $1e - 4$ , using Adam optimizer, and a batch size of 2 on randomly cropped gray scale patches of resolution  $20 \times 160 \times 160$  (11 iterations per epoch). Learning rate is halved at 10.8K, 16.2K and 22K epochs.

## References

- [1] Miao Cao, Lishun Wang, Mingyu Zhu, and Xin Yuan. Hybrid cnn-transformer architecture for efficient large-scale video snapshot compressive imaging. *International Journal of Computer Vision*, pages 1–20, 2024. [1](#), [2](#)
- [2] Prateek Chennuri, Yiheng Chi, Enze Jiang, GM Dilshan Godaliyadda, Abhiram Gnanasambandam, Hamid R Sheikh, Istvan Gyongy, and Stanley H Chan. Quanta video restoration. In *European Conference on Computer Vision*, pages 152–171. Springer, 2024. [1](#)
- [3] Dasong Li, Xiaoyu Shi, Yi Zhang, Ka Chun Cheung, Simon See, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. A simple baseline for video restoration with grouped spatial-temporal shift. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9822–9832, 2023. [2](#), [3](#)
- [4] Lydia Lindner, Alexander Effland, Filip Ilic, Thomas Pock, and Erich Kobler. Lightweight video denoising using aggregated shifted window attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 351–360, 2023. [1](#), [2](#)
- [5] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. [2](#)
- [6] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14489–14498, 2021. [1](#)
- [7] Ping Wang, Lishun Wang, and Xin Yuan. Deep optics for video snapshot compressive imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10646–10656, 2023. [2](#), [3](#)