Supplementary Material for "BIRNAT: Bidirectional Recurrent Neural Networks with Adversarial Training for Video Snapshot Compressive Imaging"

Ziheng Cheng[1][0000−0002−7504−197X], Ruiying Lu[1][0000−0002−8825−6064], Zhengjue Wang[1][0000−0002−1846−495X], Hao Zhang[1][0000−0002−2928−2692], Bo Chen[1⋆][0000−0001−5151−9388], Ziyi Meng[2,3][0000−0001−8294−8847], and Xin Yuan[4⋆][0000−0002−8311−7524]

[1] National Laboratory of Radar Signal Processing, Xidian University, Xi'an, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
[3] New Jersey Institute of Technology, NJ, USA [4] Nokia Bell Labs, NJ, USA
zhcheng@stu.xidian.edu.cn,
{ruiyinglu_xidian,zhengjuewang,zhanghao_xidian}@163.com,
bchen@mail.xidian.edu.cn,mengziyi@bupt.edu.cn,xyuan@bell-labs.com

1 Experiment Settings

The details of network architecture for BIRNAT is shown in Fig.1 (Fig. 3 in the main paper, reproduced here) and Table 1, where the output size corresponds to a mini-batch, and the activation function is LeakyReLU. As displayed in the table, the '*conv.*' refers to the convolutional operation and '*deconv.*' denotes the deconvolutional operations, and a '*resblock*' represents three two-layer convolutional layers with three residual connections. Moreover, the architecture of forward RNN is the same as backward RNN, while the parameters are not shared.

The details of our repurposed U-net architecture in the experiment is shown in Fig. 2.

⋆ corresponding author.

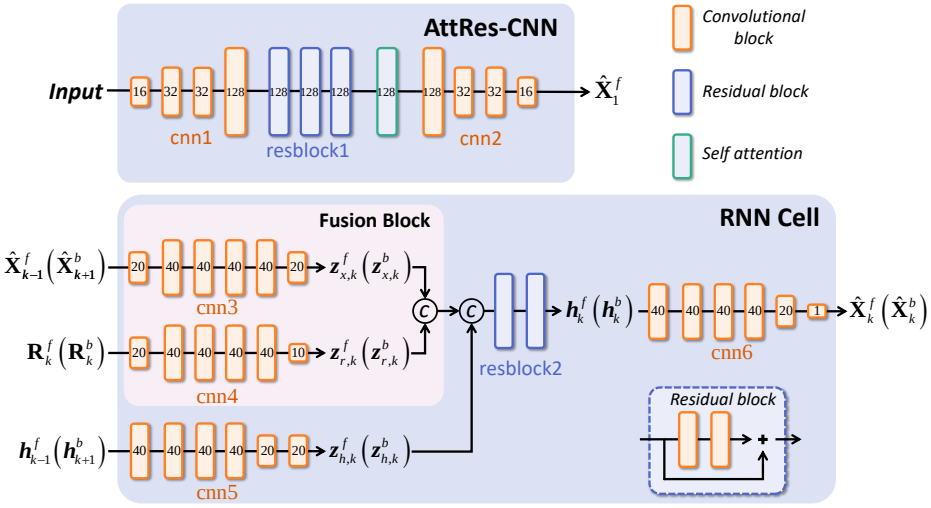

Fig. 1. Details of BIRNAT. C denotes concatenation along the 3^{rd} (temporal) dimension. The numbers in the AttRes-CNN and RNN cell denote the numbers of channels in each feature map.

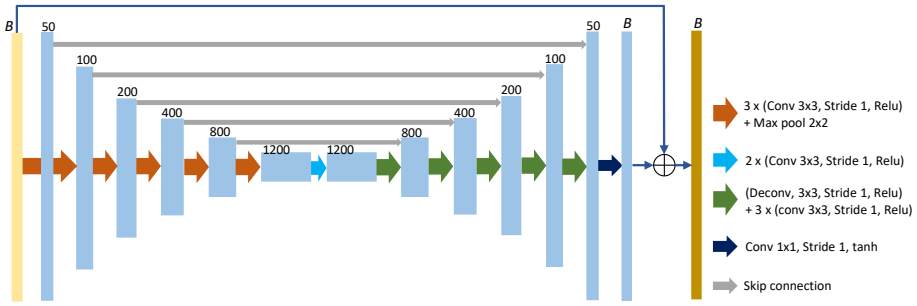

Fig. 2. The network architecture of the U-net in our experiments.

Table 1. Network architecture for the BIRNAT

Module	Operation	Kernel	Stride	Output Size
cnn1	conv.	5×5	1	$256 \times 256 \times 16$
	conv.	3×3	1	$256 \times 256 \times 32$
	conv.	1×1	1	$256 \times 256 \times 32$
	conv.	3×3	2	$128 \times 128 \times 128$
cnn2	deconv.	3×3	2	$256 \times 256 \times 32$
	conv.	3×3	1	$256 \times 256 \times 32$
	conv.	1×1	1	$256 \times 256 \times 16$
	conv.	3×3	1	$256 \times 256 \times 1$
cnn3	conv.	5×5	1	$256 \times 256 \times 20$
	conv.	1×1	1	$256 \times 256 \times 40$
	conv.	3×3	1	$256 \times 256 \times 40$
	conv.	1×1	1	$256 \times 256 \times 40$
	conv.	3×3	1	$256 \times 256 \times 40$
	conv.	3×3	1	$256 \times 256 \times 20$
cnn4	conv.	5×5	1	$256 \times 256 \times 20$
	conv.	1×1	1	$256 \times 256 \times 40$
	conv.	3×3	2	$128 \times 128 \times 40$
	conv.	1×1	1	$128 \times 128 \times 40$
	conv.	3×3	1	$128 \times 128 \times 40$
	deconv.	3×3	2	$256 \times 256 \times 10$
cnn5	conv.	3×3	1	$256 \times 256 \times 40$
	conv.	1×1	1	$256 \times 256 \times 40$
	conv.	3×3	1	$256 \times 256 \times 40$
	conv.	1×1	1	$256 \times 256 \times 40$
	conv.	3×3	1	$256 \times 256 \times 20$
	conv.	3×3	1	$256 \times 256 \times 20$
cnn6	conv.	3×3	1	$256 \times 256 \times 40$
	conv.	1×1	1	$256 \times 256 \times 40$
	conv.	3×3	1	$256 \times 256 \times 40$
	conv.	1×1	1	$256 \times 256 \times 40$
	conv.	3×3	1	$256 \times 256 \times 20$
	conv.	1×1	1	$256 \times 256 \times 1$
resblock1	$3 \times resblock$	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 3$	1	$128 \times 128 \times 128$
resblock2	$2 \times resblock$	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 3$	1	$256 \times 256 \times 50$
self-attention	$\boldsymbol{w}_1$	1×1	1	$128 \times 128 \times 16$
	$\boldsymbol{w}_2$	1×1	1	$128 \times 128 \times 16$
	$\boldsymbol{w}_3$	1×1	1	$128 \times 128 \times 128$

2 More Experimental Results

We show additional experimental results of the proposed framework on the simulated data and real data captured by SCI cameras. As a further illustration of the results shown in the main paper, we show the whole reconstructed frames of `Kobe`, `Traffic`, `Runner`, `Drop`, `Aerial` and `Vehicle` using four different methods compared with ground truth from Fig.3 to Fig.24.

We also show the reconstructed images using four different methods on the real data `Wheel`, `Handlen`, `Duominuo` and `Water Balloon` from Fig.25 to Fig.28.

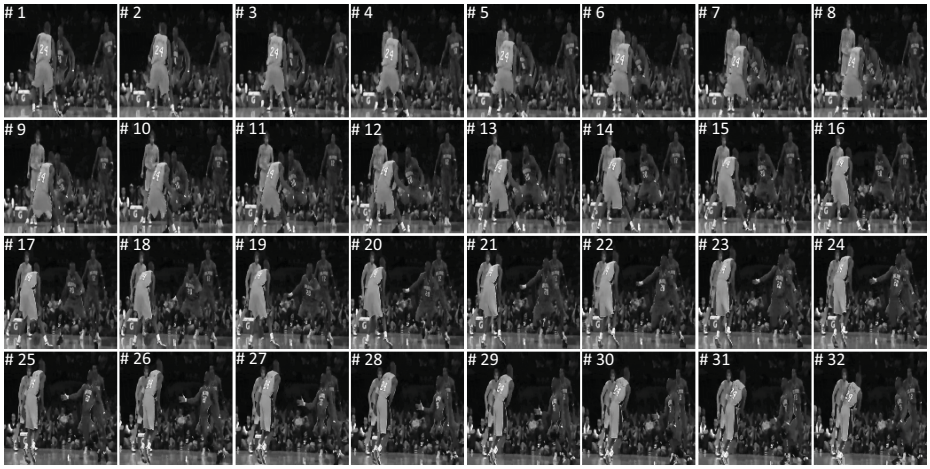

Fig. 3. Ground truth for `Kobe`.

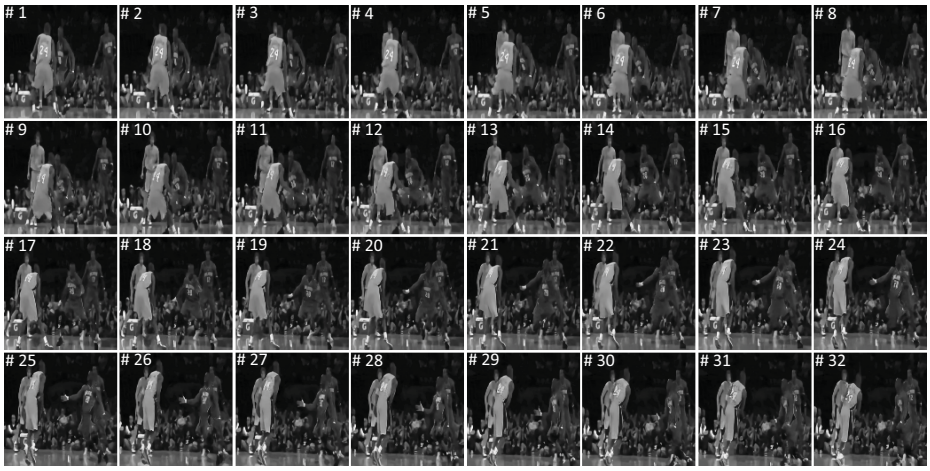

Fig. 4. Reconstructed results by our proposed BIRNAT for `Kobe`, PSNR: 32.71, SSIM: 0.9316.

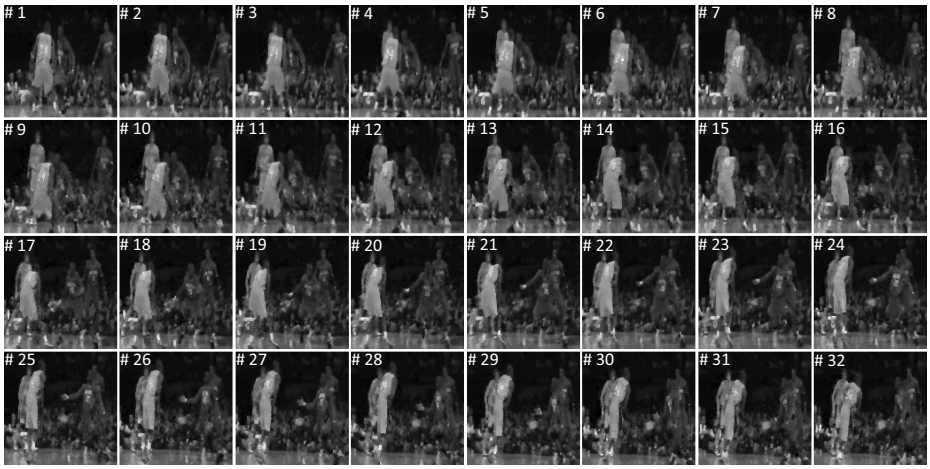

Fig. 5. Reconstructed results by GAP-TV for `Kobe`, PSNR: 26.45, SSIM: 0.8448.

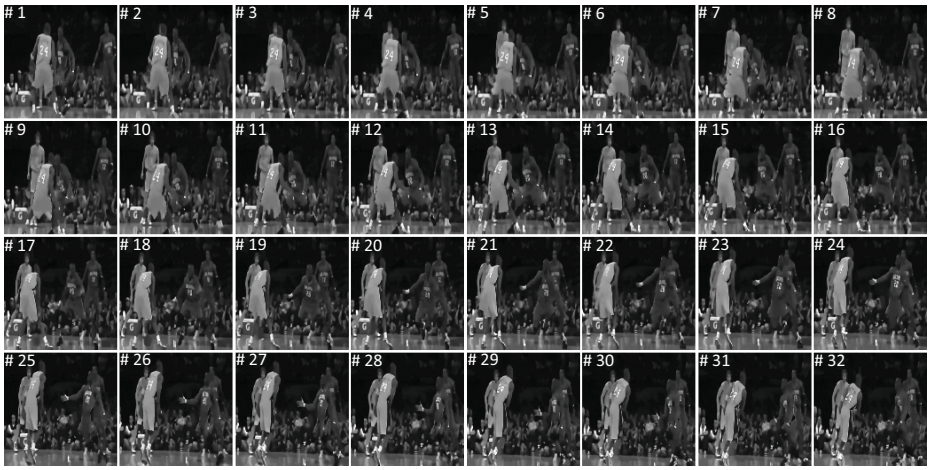

Fig. 6. Reconstructed results by DeSCI for `Kobe`, PSNR: 33.25, SSIM: 0.9518.

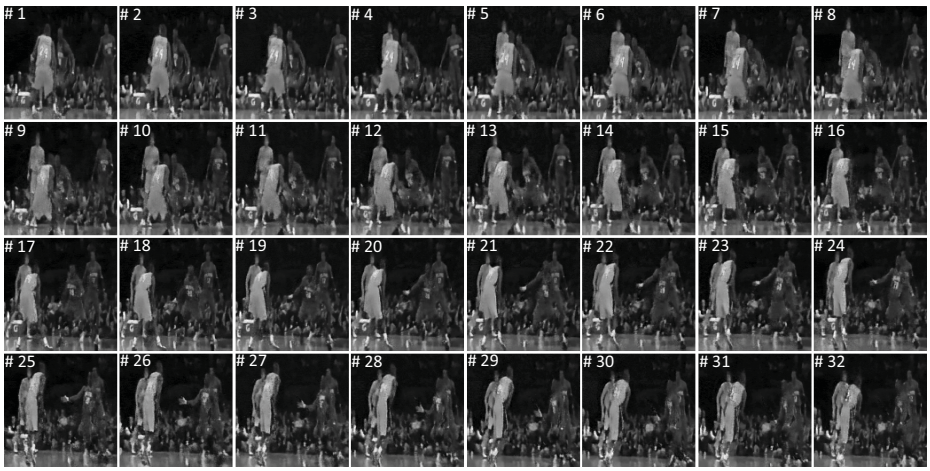

Fig. 7. Reconstructed results by U-net for `Kobe`, PSNR: 27.79, SSIM: 0.8071.

Fig. 8. Ground truth for `Traffic`.

Fig. 9. Reconstructed results by our proposed BIRNAT for `Traffic`, PSNR: 29.33, SSIM: 0.9322.

Fig. 10. Reconstructed results by GAP-TV for `Traffic`, PSNR: 20.89, SSIM: 0.7148.

Fig. 11. Reconstructed results by DeSCI for `Traffic`, PSNR: 28.72, SSIM: 0.9250.

Fig. 12. Reconstructed results by U-net for `Traffic`, PSNR: 24.62, SSIM: 0.8403.

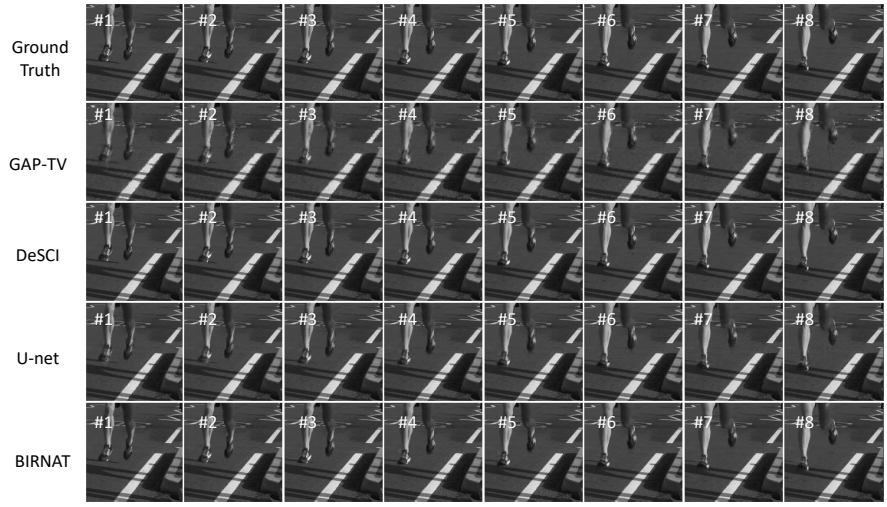

Fig. 13. Ground truth and reconstructed results by different methods on `Runner`. PSNR and SSIM: 28.81, 0.9092 (GAP-TV); 38.76, 0.9693 (DeSCI); 34.12, 0.9471 (U-net); 38.70, 0.9685 (our proposed BIRNAT).

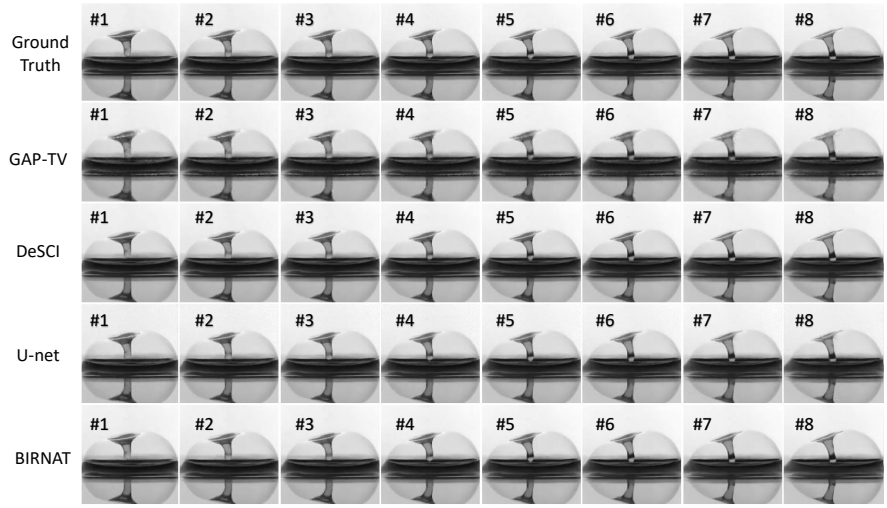

Fig. 14. Ground truth and reconstructed results by different methods on `Drop`, PSNR and SSIM: 34.74, 0.9704 (GAP-TV); 43.22, 0.9925 (DeSCI); 36.56, 0.9494 (U-net); 42.28, 0.9896 (our proposed BIRNAT).

Fig. 15. Ground truth for `Aerial`.

Fig. 16. Reconstructed results by our proposed BIRNAT for `Aerial`, PSNR: 28.99, SSIM: 0.9136.

Fig. 17. Reconstructed results by GAP-TV for `Aerial`, PSNR: 25.05, SSIM: 0.8281.

Fig. 18. Reconstructed results by DeSCI for `Aerial`, PSNR: 25.33, SSIM: 0.8603.

Fig. 19. Reconstructed results by U-net for `Aerial`, PSNR: 27.18, SSIM: 0.8690.

Fig. 20. Ground truth for `Vehicle`.

Fig. 21. Reconstructed results by our proposed BIRNAT for `Vehicle`, PSNR: 27.84, SSIM: 0.9092.

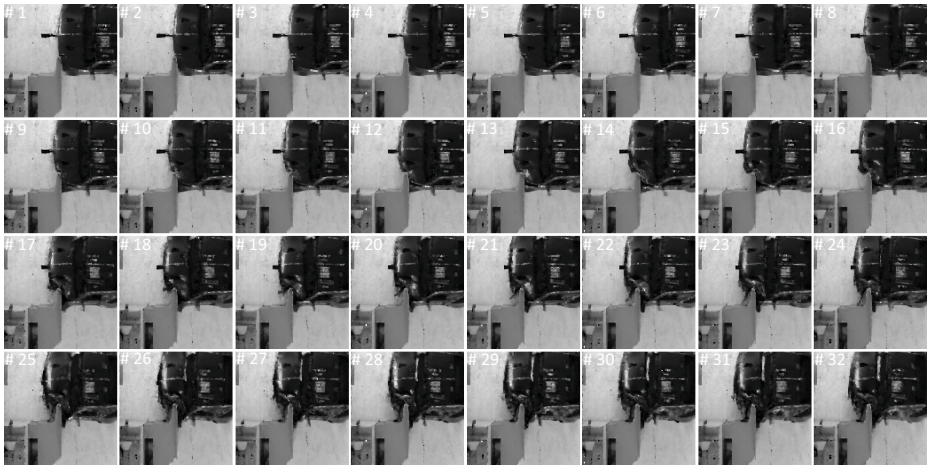

Fig. 22. Reconstructed results by GAP-TV for `Vehicle`, PSNR: 24.82, SSIM: 0.8383.

Fig. 23. Reconstructed results by DeSCI for `Vehicle`, PSNR: 27.04, SSIM: 0.9094.

Fig. 24. Reconstructed results by U-net for `Vehicle`, PSNR: 26.43, SSIM: 0.8817.

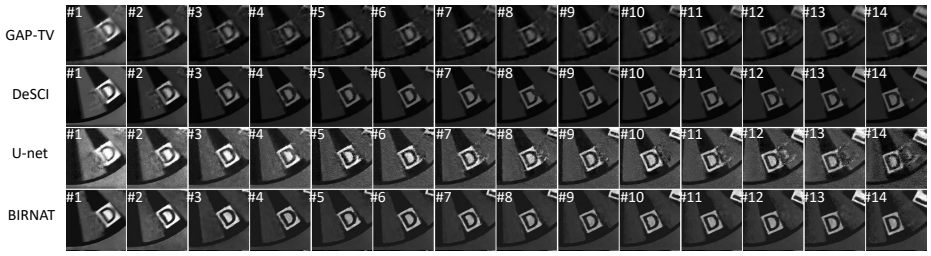

Fig. 25. The results of different methods on real data `Wheel`.

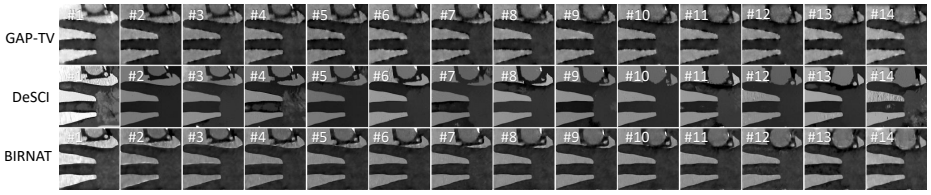

Fig. 26. The results of different methods on real data `Handle`.

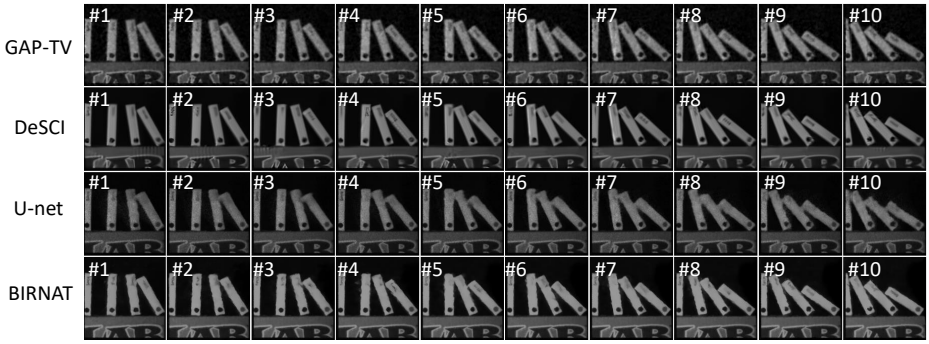

Fig. 27. The results of different methods on real data for `Domino`.

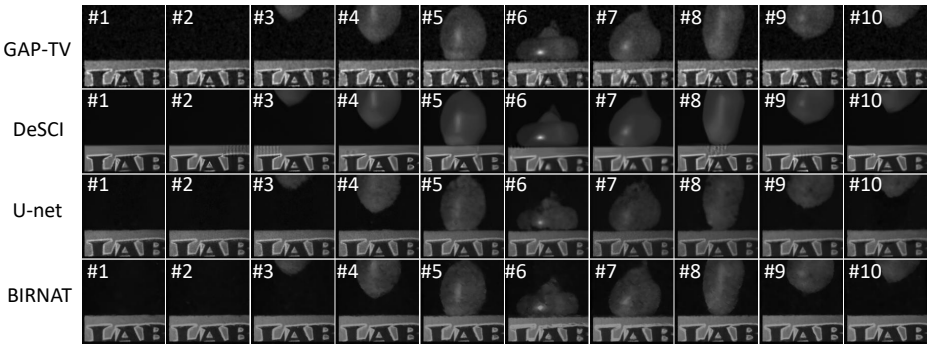

Fig. 28. The results of different methods on real data `Water Balloon`.