

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073

# COMISR: Compression-Informed Video Super-Resolution

## Supplementary Material

Anonymous ICCV submission

Paper ID 10703

### 1. Model Details

The overall COMISR model is shown in the Figure 2 in the paper manuscript. Here we present more details of the two modules: detail-preserving flow estimation and HR frame generate.

#### 1.1. Detail-Preserving Flow Estimation

The flow is estimated via a network architecture as is shown in Figure 1. At each training mini-batch, a short video clip (e.g. 7 frames) is used for training. The flow estimation can be divided into two parts. The first part is to learn the motion discontinuities between the consecutive frames. The second part is for upscaling the estimated flow, which will be then used on the HR frames. The upscaling process is designed by a learn residual added to a  $4 \times$  bilinear upsampling. Such residual is implemented by repeating a  $2 \times$  transpose convolutional layer twice.

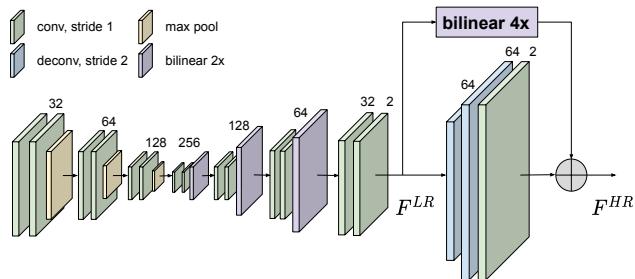


Figure 1. Detail-Preserving Flow Estimation

All the convolutional kernels are  $3 \times 3$ . The number of filters in each layer are marked in the Figure 1. The output flow estimation is used to warp the  $t - 1$  generated HR frame to the  $t$  timestamp, and then used for generating the HR frame in the  $t$  timestamp.

#### 1.2. HR Frame Generator

As shown in Figure 2 of the manuscript, the input of the HR frame generator is the concatenation of a space-to-depth results of the current estimated HR frame and the

current input LR frame. In the HR frame generator, 10 repeated residual blocks are first employed to extract high-level features. Then a upscaling module, similar to *Detail-Preserving Flow Estimation* in Section 1.1 is used to create estimated HR frame.

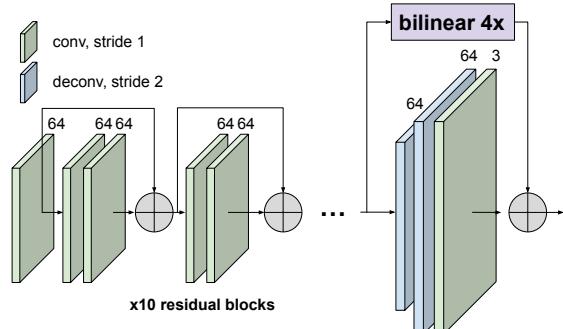


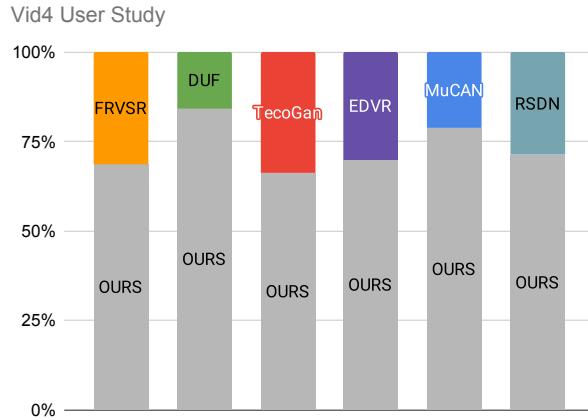
Figure 2. HR Frame Generator

### 2. User study

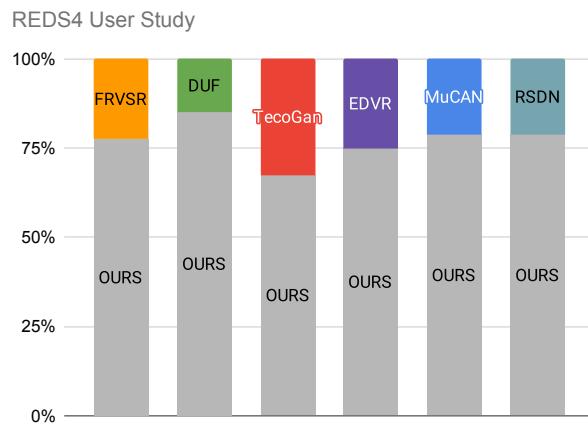
To better evaluate the visual quality of the generate HR videos, we conduct a user study using Amazon MTurk [2] on the Vid4 [4], REDS4 [5], and Vimeo90K [8] testing dataset. We evaluate the COMISR model against all other methods using videos compressed with CRF25<sup>1</sup>. In each test, two videos generated by the COMISR model and other methods are presented side by side<sup>2</sup>, and each user is asked “which video looks better?” For the Vid4 and REDS4 datasets, all the videos are used for the user study. For the Vimeo90k testing set, the first 20 videos are used. For each of the video pairs, we assign to 20 different raters. The raters are selected with 90%+ HIT approval rate, and with 500+ HITs selected to participate in this user study. For the Vimeo90k testing set, we are able to evaluate all testing videos on FRVSR, TecoGan, EDVR and RSDN. However, as the EDVR method only outputs one middle frame, it is

<sup>1</sup>The default CRF value for the open source code ffmpeg is 23.<sup>2</sup>The left and right videos are randomly shuffled to avoid any bias.

108 not included in the user study. All the results are summa-  
 109 rized in Figure 3, 4, and 5.  
 110



127 Figure 3. Vid4 user study results. All four sequences are used.  
 128 The comparison is between COMISR and all other methods. Re-  
 129 sults show that users favored COMISR against all other compared  
 130 methods.

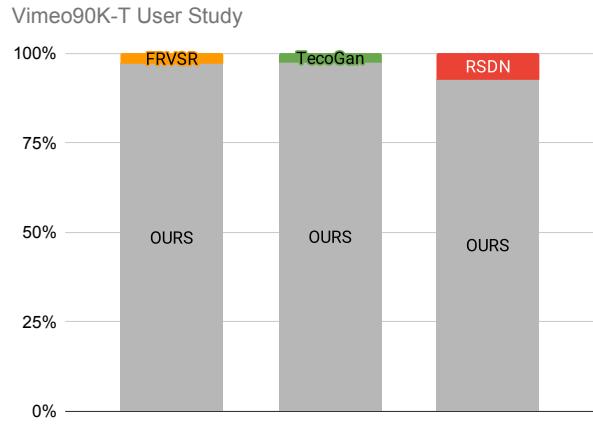


147 Figure 4. REDS4 user study results. All four sequences are used.  
 148 The comparison is between COMISR and all other methods. Re-  
 149 sults show that users favored COMISR against all other compared  
 150 methods.

### 3. Experimental Results

#### 3.1. Vimeo-90K-T dataset

156 We evaluate the COMISR model against the state-of-the-  
 157 art VSR methods on the Vimeo-90K-T dataset [8], includ-  
 158 ing FRVSR [6], EDVR [7], TecoGan [1], and RSDN [3].  
 159 The Vimeo-90K-T dataset contains 7824 short video clips,  
 160 where each clip only has 7 frames. Similar to the observa-  
 161 tion in [3], the recurrent-based method may not take full ad-



170 Figure 5. Vimeo90k testing set user study results. The first 20 se-  
 171 quences are used. The comparison is between COMISR and three other  
 172 methods. Results show that users greatly favored COMISR  
 173 against all other compared methods.

177 vantages due to very short video clips, the COMISR model  
 178 can still outperform others on the compressed videos. We  
 179 show both quantitative result below, and qualitative results  
 180 in Section 3.2.

	Uncompressed	CRF25
FRVSR [6]	35.64 / 0.932	30.07 / 0.788
TecoGan [1]	34.07 / 0.909	29.84 / 0.784
EDVR [7]	37.61 / 0.949	30.53 / 0.844
RSDN [3]	37.23 / 0.947	29.63 / 0.815
Ours	35.71 / 0.926	31.05 / 0.816

192 Table 1. Performance evaluation of Y-channel on the Vimeo90K  
 193 testing set.

#### 3.2. Qualitative Comparisons

196 We add additional visual comparison results from the  
 197 Vid4, REDS4, and Vimeo90K testing datasets. For the  
 198 Vimeo90K testing set, we are able to run the FRVSR,  
 199 EDVR, TecoGan, and RSDN on all the testing videos. Note  
 200 that EDVR only generates the middle HR frame (#4) out of  
 201 the 7 input LR frames, so the comparison on the Vimeo90K  
 202 testing set is on the middle frame for all the methods.

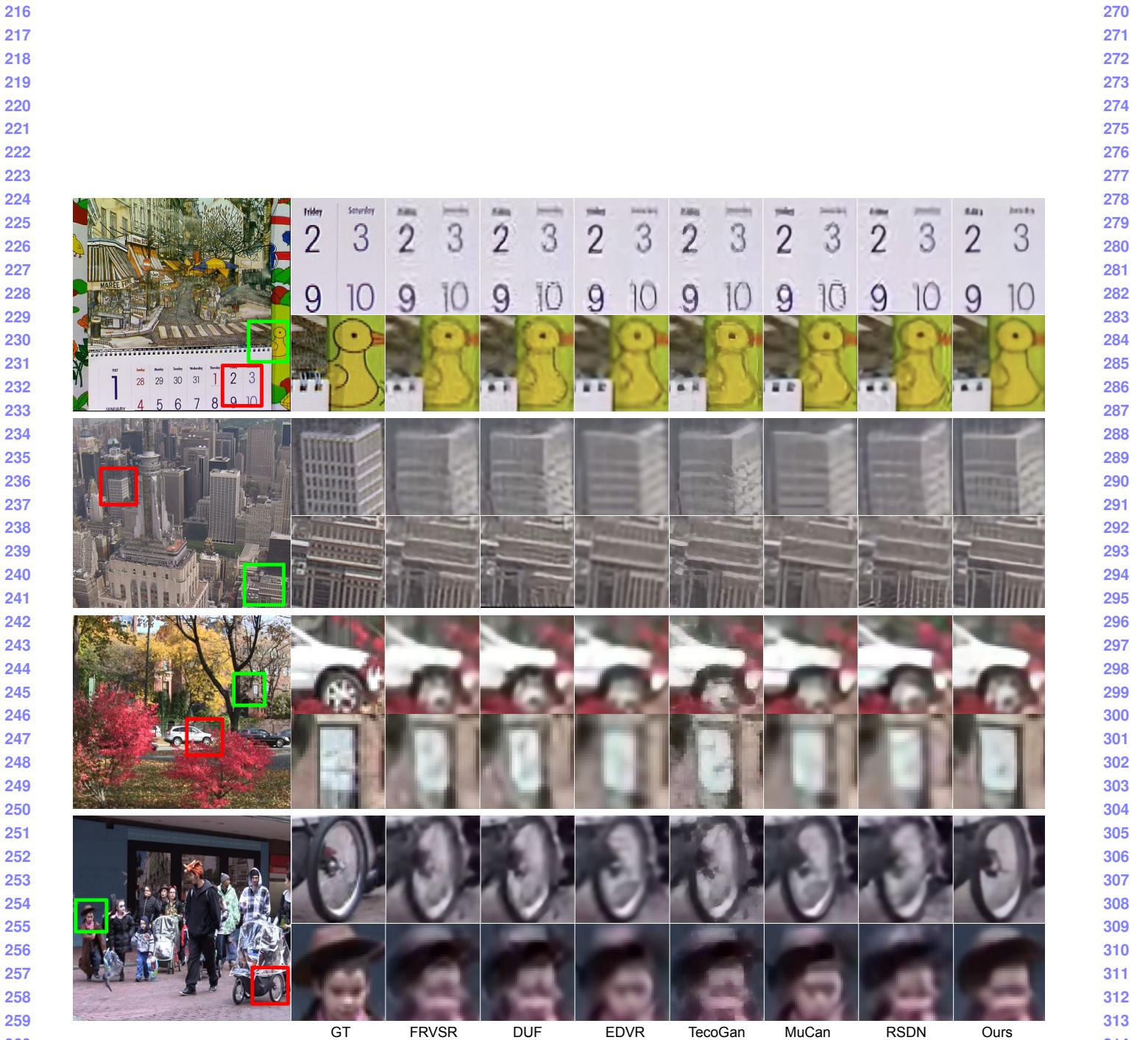


Figure 6. Visual example of the Vid4 dataset. All the LR input frames are compressed with CRF25. Zoom in for best view.

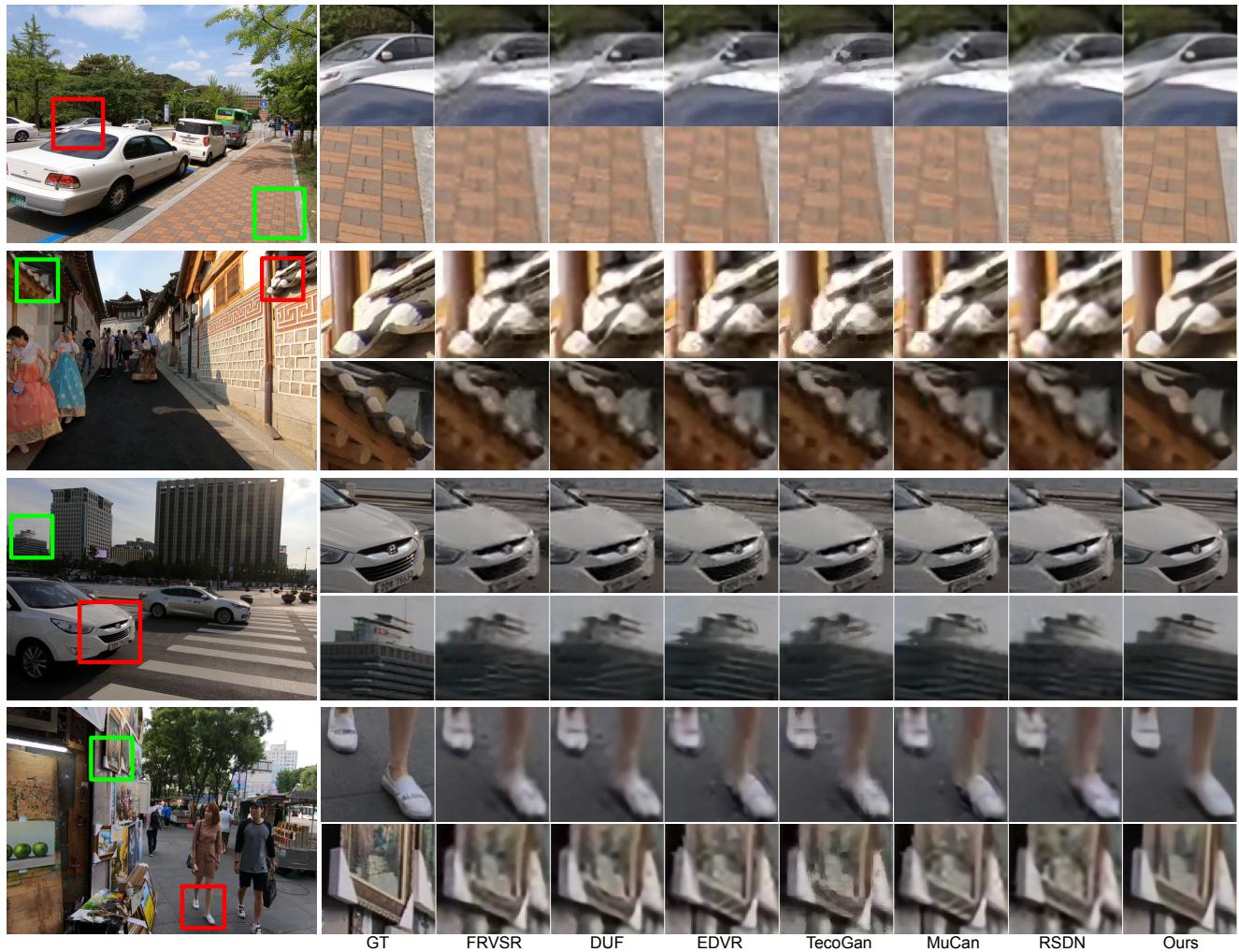


Figure 7. Visual example of the REDS4 dataset. All the LR input frames are compressed with CRF25. Zoom in for best view.

324 378  
325 379  
326 380  
327 381  
328 382  
329 383  
330 384  
331 385  
332 386  
333 387  
334 388  
335 389  
336 390  
337 391  
338 392  
339 393  
340 394  
341 395  
342 396  
343 397  
344 398  
345 399  
346 400  
347 401  
348 402  
349 403  
350 404  
351 405  
352 406  
353 407  
354 408  
355 409  
356 410  
357 411  
358 412  
359 413  
360 414  
361 415  
362 416  
363 417  
364 418  
365 419  
366 420  
367 421  
368 422  
369 423  
370 424  
371 425  
372 426  
373 427  
374 428  
375 429  
376 430  
377 431

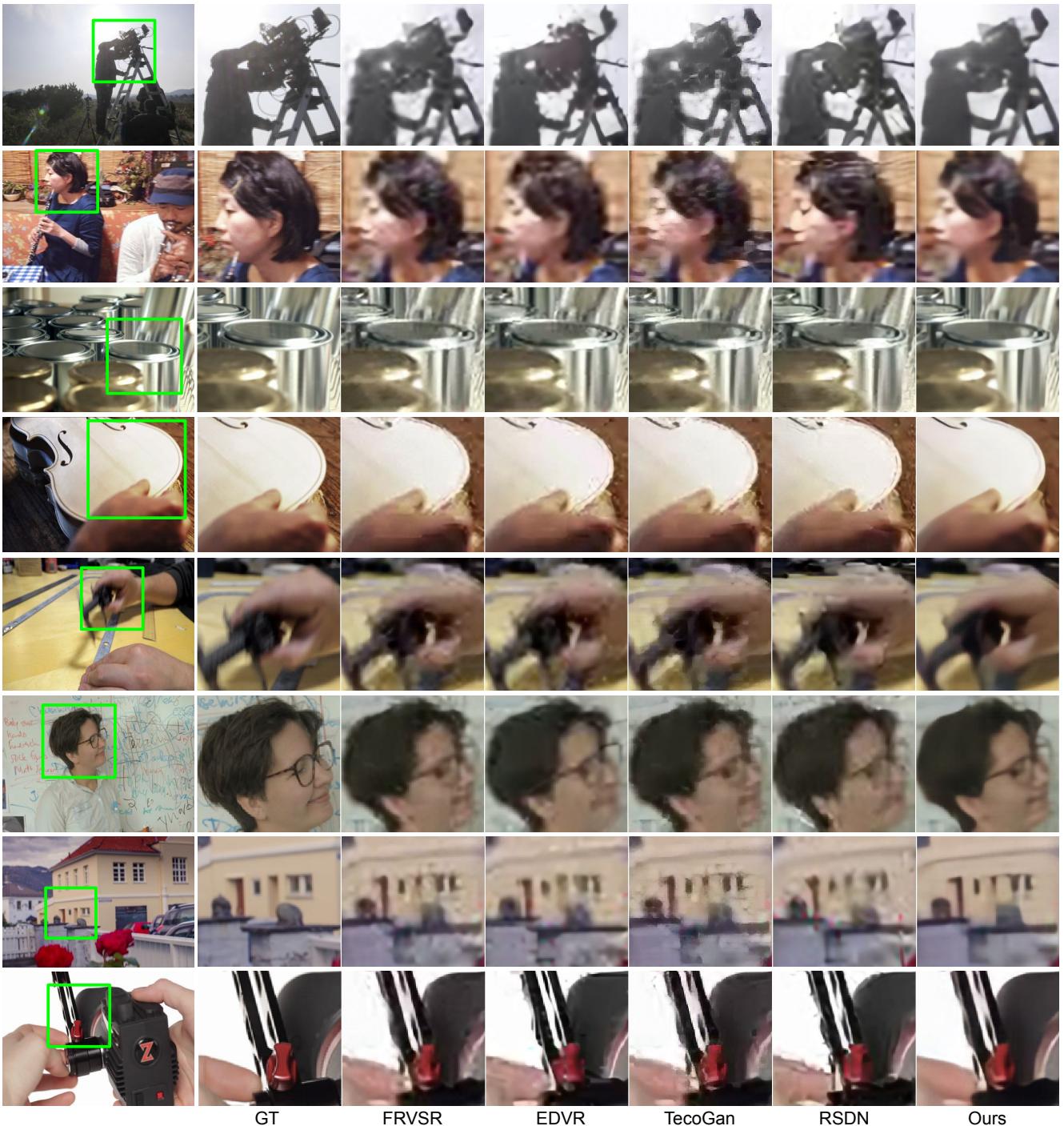


Figure 8. Visual example of the Vimeo90k testing set. All the LR input frames are compressed with CRF25. Zoom in for best view.

540 **References**

- 541  
542 [1] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixe,  
543 and Nils Thuerey. Learning temporal coherence via self-  
544 supervision for gan-based video generation. *ACM Transac-  
545 tions on Graphics*, 2018. 2  
546 [2] Amazon Inc. Amazon mturk. <https://www.mturk.com/>, 2021. 1  
547 [3] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin  
548 Wang, and Qi Tian. Video super-resolution with recurrent  
549 structure-detail network. In *ECCV*, 2020. 2  
550 [4] C. Liu and D. Sun. A Bayesian approach to adaptive video  
551 super resolution. In *CVPR*, 2011. 1  
552 [5] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik  
553 Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu  
554 Lee. Ntire 2019 challenge on video deblurring and super-  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593 resolution: Dataset and study. In *CVPR Workshops*, 2019.  
594  
1  
595 [6] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew  
596 Brown. Frame-Recurrent Video Super-Resolution. In *CVPR*,  
597 2018. 2  
598 [7] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and  
599 Chen Change Loy. Edvr: Video restoration with enhanced de-  
600 formable convolutional networks. In *CVPR Workshops*, 2019.  
601  
2  
602 [8] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and  
603 William T Freeman. Video enhancement with task-oriented  
604 flow. *International Journal of Computer Vision*, 2019. 1, 2  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647