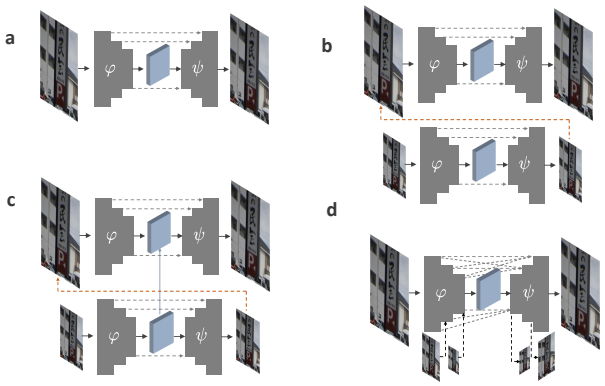
Single Image Deblurring with Row-dependent Blur Magnitude —Supplemental Material—

Xiang Ji[1] Zhixiang Wang[1,2] Shin'ichi Satoh[2,1] Yinqiang Zheng[1†]
[1]The University of Tokyo, Japan [2]National Institute of Informatics, Japan
{jixiang,wangzhixiang}@g.ecc.u-tokyo.ac.jp, satoh@nii.ac.jp, yqzheng@ai.u-tokyo.ac.jp

In this supplemental material, we present implementation details (Section A) and additional results (Section B.3) to complement the main manuscript.

A. Implementation Details



Extended Figure S.1: Comparison of coarse-to-fine image deblurring network architectures: (a) basic U-net, (b) multi-scale U-net, (c) scale-recurrent U-net, (d) multi input-output single U-net.

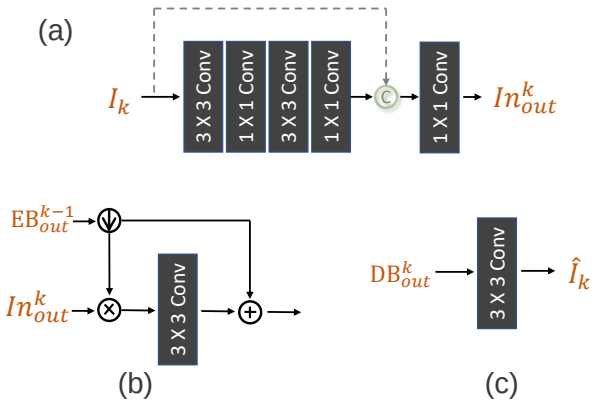
A.1. Architecture Design

To explore efficient architecture of models for GRR image deblurrig, we revisit the vanilla coarse-to-fine strategy and summarize representative backbones of networks as shown in **Extended Figure S.1**. Basic U-net structure [18] greatly increases regression ability and is widely used in recent work of FlowNet [3], video super-resolution [20], view synthesis [10], etc. Improvements have also been made in image deblurring by using this U-shape structure [11]. The network first progressively transforms the input image into feature maps with low resolution and more channels (encoder), then converts them back to image space (decoder). Skip-connections between corresponding feature maps with the same scale promote information propagation and accelerate convergence. However, single U-net is hard to handle complex motions with different scales. PSS-NSC [4] uses multiple stacks of sub-networks (**Extended Figure S.1**b) with different resolution copies of the image as input. And the resultant output from a coarser sub-network is concatenated with the input of a finer sub-network to enable processing nonuniform blur. But, multi-stacks of sub-networks will increase the number of parameters quickly and affect convergence process negatively. SRN [21] adopts a novel recurrent structure across multi-scale sub-networks (**Extended Figure S.1**c). They propose sharing network weights of sub-networks to reduce training difficulty and introduce stability benefits. Similar strategy is also adopted in [15]. Recently, Cho *et al*. [2] mimic multi-cascaded sub-networks and significantly ease the training difficulty by using a single U-net with multi input and multi output (**Extended Figure S.1**d).

On the other hand, Transformers have shown a huge performance gains on image restoration by relying on its global dependency modeling and flexible attention mechanism. Researchers have explored an alternative way to combine coarse-to-fine structure with Transformer block. Uformer [23] build their model based on basic U-net structure and modify the convolution layers to Transformer blocks, achieving superior performance on restoring details while introducing marginal extra parameters and computational cost. Inspired by this, we take a further step to integrate the multi input-output U-shaped backbone with our novel RSS-T block for the targeted single GRR debluring. The core designs of our RSS-T model have been illustrated in Figure 2–3 in the main paper. The other modules are shown in **Extended Figure S.2**. Similar to [2], we exploit an shallow convolution blocks to extract feature of input image and then merge them with the output of previous encoder by feature attention module (FAM). The up and down sampling operations in Figure 2 are implemented using 4×4 transposed convolution and 3×3 convolution with stride of 2, respectively. Furthermore, we use 1×1 convolutional layer followed by LeakyReLU in decoder block to fuse features from all encoder blocks and previous decoder. The final

†Corresponding author

output projections with different scales are implemented by a single 3×3 convolution layer.



Extended Figure S.2: The structures of left sub-modules: (a) input projection module, (b) feature attention, and (c) output projection module.

A.2. Data Acquisition

A.2.1 Camera systems

To facilitate the development and evaluation of GRR deblurring, we take paired GRR/GS videos to offer a new dataset captured under urban scenes named GRR-real. Learning-based methods for debluring or RS correction, usually synthesize required data from high frame-rate GS videos. But real datasets directly captured by cameras are also essential for training and evaluation. Following [26, 1, 24], we construct our camera system for capturing GRR/GS pairs. In the system, the GS and GRR cameras are attached to a beam-spliter followed by a relay lens and an objective lens. Because the relay lens will turn image upside down, we install GRR camera inversely to offset this effect.

As for alignment, we manually tune the poses of two cameras to make sure their fist scanlines are aligned. We also set the exposure time of the first scanline of GRR camera identical to that of GS camera, and reset them simultaneously, which is controlled by a signal generator. Since pixel sizes of the two cameras are not completely the same, we adjust the GS images through a precalibrated homography. Besides the camera system, we also use another GRR camera with different settings to capture data for our generalization evaluation in Section 4.1. All detailed information is illustrated in **Extended Figure S.3** and **Extended Figure S.4** .

A.2.2 Real-world dataset

Our GRR-real dataset consists of 64 video sequences taken from different urban scenes with a resolution of 640×480, including streets, driveways, buildings, trees, vehicles, and so on. Each sequence contains 256 frames with GRR version and corresponding ground truth. We split the dataset into a training set with 50 sequences and validating and testing sets with 7 sequences respectively.

During capturing process, we not only considered the dynamic objects in the scene but also tried to move our camera system with varying rotation and translation. We show some selected samples of our dataset in **Extended Figure S.5**. In the difference map, larger differences are highlighted in black while smaller differences are in gray. We could clearly find that the blur and brightness of our GRR data is highly correlated to image rows which is quite different with row-independent blur of GS images.

A.2.3 Synthesized dataset

Although, we collected real-world data to perform single GRR correction. As complementary part, we also validated the advantage of GRR over RS mode on synthetic data. Following the convention [14, 21, 2], synthesis process is also grounded on GOPRO data [12] consisting of 33 videos with resolution of 1280×720. Each video clip contains about 1200 consecutive frames at 240fps. For benefits of generating more realistic effects, GOPRO is firstly interpolated at $\times 64$ using an off-the-shelf video interpolation algorithm [5].

As discussed formulation of three modes in Introduction part, The RS videos are synthesized by sequentially copying a row of pixels from high-speed videos and blur generated by averaging them, as in previous works [8, 21]. The GRR synthesizing process is similar to that of bur, but has two different parts: 1) Different rows of an GRR frame are contributed by variant numbers of high-speed frames; 2) All used high-framerate sharp frames except for the first one are multiplied by a factor δ, which determines the ratio between readout time and the first scanline's exposure duration. The generation process of RS, GRR and blur videos are strictly aligned to each frame ensuring they capture identical contents of the scene. In practice, we centrally crop frames to 512, and set $N = 512, \delta = 0.001$. The stride of synthesizing process is set as same as N. Finally, we have 33 videos with three degradation counterparts and corresponding GT frames.

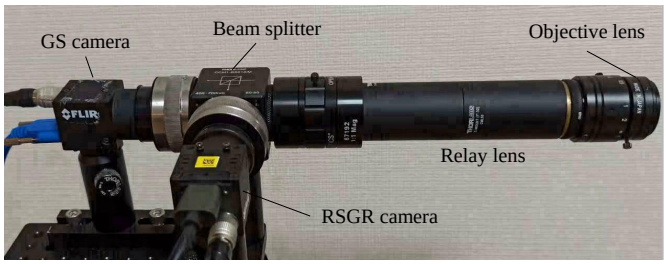
A.3. Training and evaluation

A.3.1 Training

Loss. We train our network using the multi-scale Charbonnier loss (MSC) [25] to reconstruct clear latent images:

$$L_{MSC} = \sum_{k=0}^{K} \frac{1}{t_k} \sqrt{\|\mathbf{I}_k - \hat{\mathbf{I}}_k\|^2 + \epsilon^2}\,. \tag{1}$$

For better restoring high-frequency component, multi-scale frequency reconstruction (MSFR) [2] loss is also presented:



Device	GRR camera	GS camera
Type	FLIR BFS-U3-63S4C	BFLY-U3-23S6C-C
Sensor	Sony IMX178	Sony IMX249
Frame rate	30 FPS	30 FPS
Resolution	640×480	640×480
1st exposure	1 ms	1 ms
Scan direction	Top-to-bottom ↓	–

Extended Figure S.3: Our synchronized and aligned dual camera system.

Device	GRR camera II
Type	EO-1312LE
Sensor	e2v EV76C560
Frame rate	30 FPS
Resolution	1280×1024
1st exposure	Auto Mode
Scan direction	Top-to-bottom ↓

Extended Figure S.4: Another GRR camera.

$$L_{MSFR} = \sum_{k=0}^{K} \frac{1}{t_k} \|\mathcal{F}(\mathbf{I}_k) - \mathcal{F}(\hat{\mathbf{I}}_k)\|_1 \,, \tag{2}$$

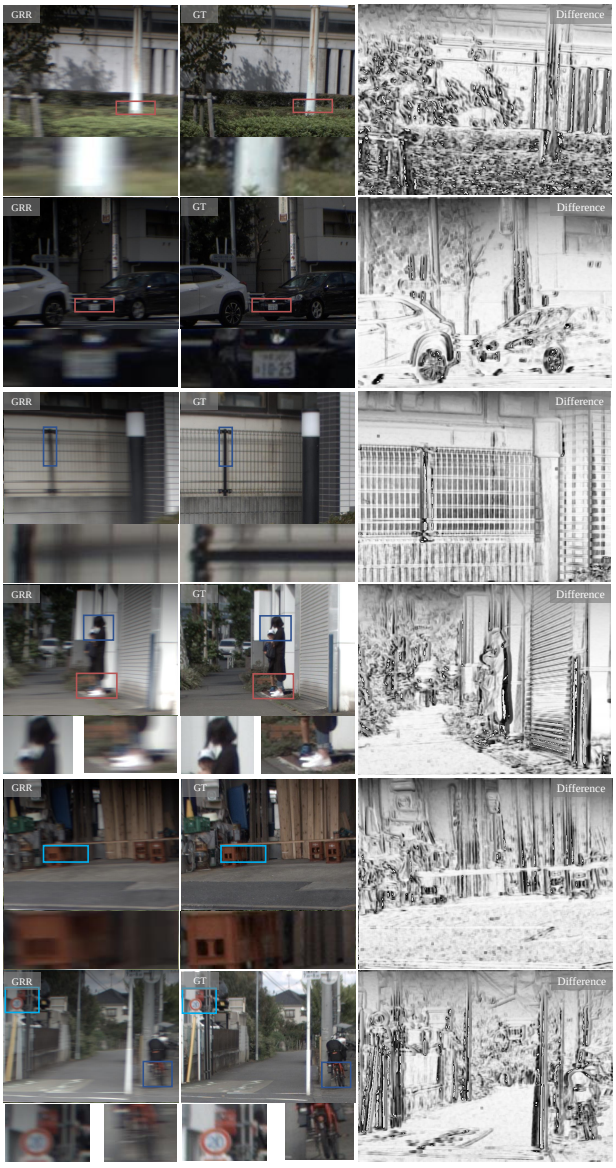
where $\mathcal{F}(\cdot)$ denotes the fast Fourier transform (FFT) that transfers image signal to the frequency domain. K is the number of scale levels and t_k is total pixels number of image on k^{th} scale level. The total loss function is given by:

$$L_{total} = L_{MSC} + \lambda L_{MSFR} \,. \tag{3}$$

We experimentally set $\lambda = 0.1$.

Training details. We implemented our network in PyTorch [16]. Following the common training strategy of Transformer, we use Adam solver [7] with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. For data augmentation, each patch was horizontally flipped and reversed in RGB channel with a probability of 0.5. Corresponding ground truth patches are generated accordingly. The network is trained in 500 epochs with a learning rate of 10^{-4} and batch size of 8. We use the cosine decay strategy to decrease the learning rate to 1e-6. The number of scale levels in our RSS-T model equals 4 by default. And the dimension of each head in Transformer block d_h equals initial channel dimension $C = 32$ by setting number of heads in each Transformer block as $\{1, 2, 4, 8, 8, 4, 2, 1\}$. Besides, number of Transformer layers of each RSS-T block in decoder or encoder is set as $\{2, 2, 2, 2, 2, 2, 2, 2\}$. Notably, when training on synthesized dataset, the epoch is set as 1000 and each frame is cropped into 256×256.

Notably, when tackling row-independent blur magnitude of GS images, researchers tend to augment their training data with multiple strategies, for example, vertical flipping, rotation with $90°, 180°$ and $270°$. Here, we do not exploit other



Extended Figure S.5: Samples from our real-world dataset.

Extended Table S.1: Experimental results of different data augmentation strategies.

Index	T	M	B	F
input	15.12/0.67	21.83/0.78	20.08/0.76	17.61/0.74
A1	25.53/0.87	25.06/0.83	21.44/0.73	22.11/0.81
A2	23.44/0.87	23.42/0.81	21.88/0.78	22.18/0.82
A3	26.59/0.91	28.61/0.88	27.31/0.84	26.49/0.88

measures to enhance our training set, because the pattern of row-dependent blur could be corrupted by those operations. To support this strategy, we also conducted related experiments as shown in **Extended Table S.1**. The *A1* (horizontal and vertical flipping), *A2* (all measures mentioned above) and *A3* (measures we chose) represent experimental results from different data augmentation settings. The large performance gaps between *A1*, *A2* with *A3* come from the destruction of GRR blur pattern and early stopping triggered by unstable training process (Note that the RSS-T model used here is the baseline version, not final one).

A.3.2 Evaluation

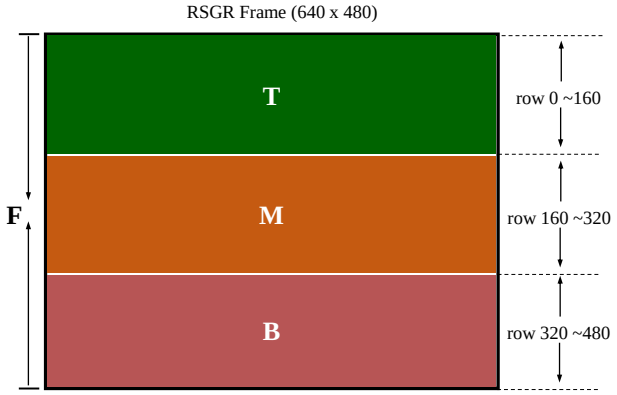
We adopt the commonly-used Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [22] metrics to evaluate the restoration performance. These metrics are calculated in the RGB color space for a single video frame. Considering the distortion is row-dependent, we also divide the video frame into three parts along vertical dimension for evaluation. Our experiments were conducted on Intel i7-8700K and two GPUs of NVIDIA GeForce RTX 3090.

Different from the training phase, we use the full video frame for testing. By convention, we use PSNR and SSIM as our metrics to evaluate the deblurred output. But considering the row-related blur and brightness of GRR images, we also presented the PSNR and SSIM of top (T), middle (M) and bottom (B) areas. The dividing method is shown in **Extended Figure S.6**. We computed the complexity of all algorithms as in the manuscript. Flops and test time are measured by deblurring one GRR image (640×480) on a NVIDIA Geforce RTX 3090 GPU. We see that the computation cost of our model is in medium level. And the test time is also lower than state-of-the-art RS correction models and Transformer-based image restoration models.

B. Extended Comparisons

B.1. Generalization evaluation.

To evaluate the generalization performance of our network, we also collect another three sequences of GRR images with different hardware and δ (*i.e.,* the ratio of readout



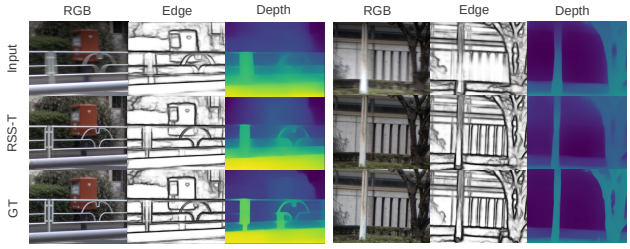
Extended Figure S.6: Dividing manner of non-overlapping patches.

Extended Figure S.7: Generalization ability. We train our model on a fixed camera setting and test on captured sequences with different hardware and δ.

time and exposure duration of first scanline). **Extended Figure S.7** demonstrates our RSS-T can successfully generalize to different data without introducing artifacts and undesired distortions.

B.2. Extended Results on Downstream Tasks

To prove our GRR deblurring approach works well on real applications. We presented experimental results of edge detection [9] and depth estimation [17] as illustrated in **Extended Figure S.8**. The results suggest that our deblurred GRR images significantly improve the detection or prediction quality, validating that our method could facilitate downstream tasks.



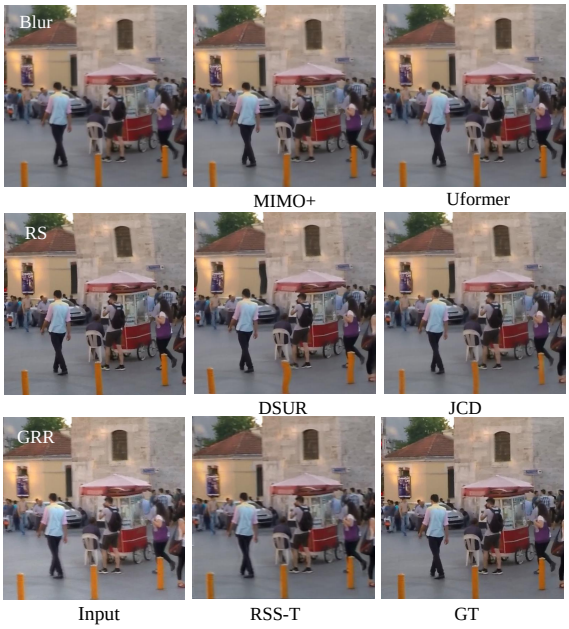
Extended Figure S.8: Downstream task performance. Predicted edge and depth by original GRR, our deblurred and GT images.

B.3. Additional Results

Qualitative comparison of ablation study and three modes are given in **Extended Figure S.9** and respectively. Visual

results of comparisons on third-party dataset [24] are in **Extended Figure S.11** and **S.12**. Supplemental qualitative results for generalization validation are in **Extended Figure S.13**. **Extended Figure S.14–S.20** are the additional results of Figure 4.

Extended Figure S.9: Qualitative results of ablation study.



Extended Figure S.10: Qualitative comparison of three modes.

References

[1] Cenek Albl, Zuzana Kukelova, Viktor Larsson, Michal Polic, Tomas Pajdla, and Konrad Schindler. From two rolling shutters to one global shutter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2505–2513, 2020. 2

[2] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4641–4650, 2021. 1, 2, 8, 9, 10, 11, 12, 13, 14

[3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1

[4] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3848–3856, 2019. 1

[5] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[6] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4038–4047, 2017. 6, 8, 9, 10, 11, 12, 13, 14

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

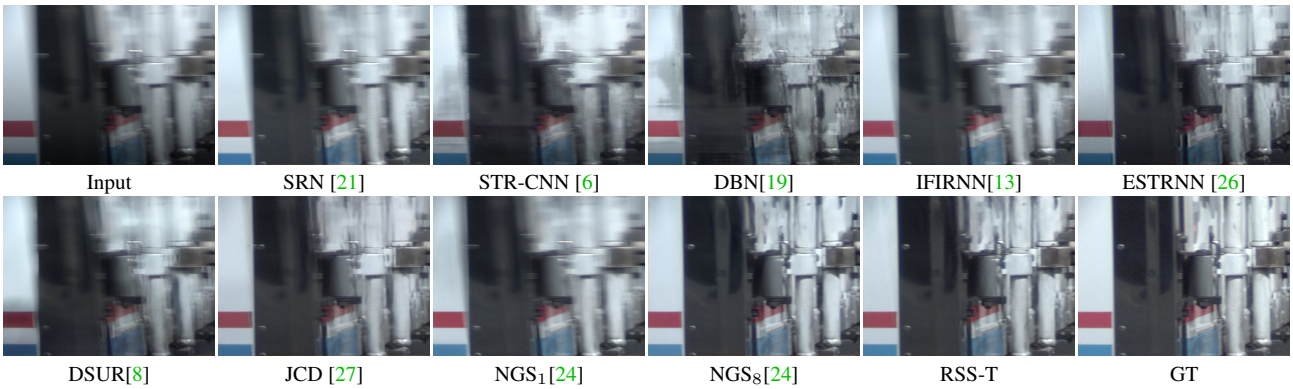
[8] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5941–5949, 2020. 2, 6, 8, 9, 10, 11, 12, 13, 14

[9] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3000–3009, 2017. 4

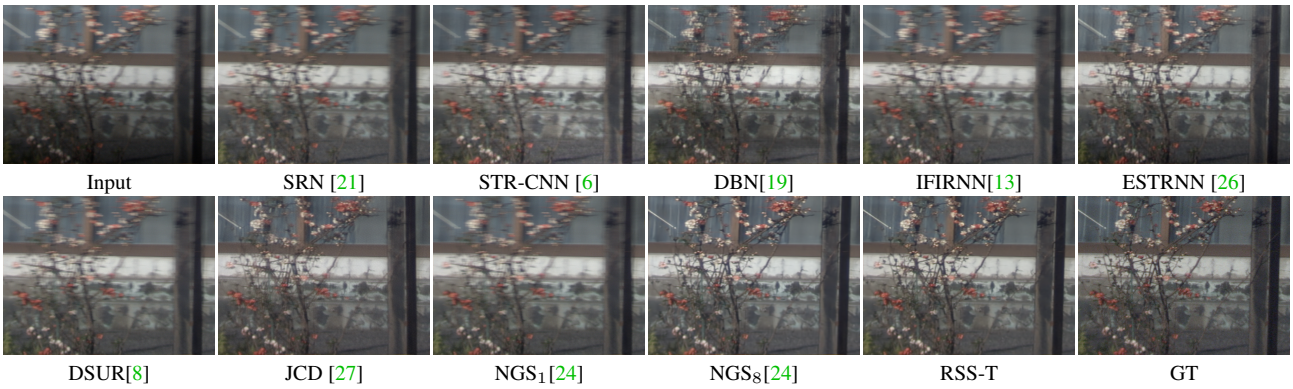
[10] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017. 1

[11] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*, 29, 2016. 1

[12] 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



Extended Figure S.11: Qualitative results on third party dataset I. We compare our method on another GRR dataset proposed in [24].



Extended Figure S.12: Qualitative results on third party dataset II. We compare our method on another GRR dataset proposed in [24].

[13] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8102–8111, 2019. 6, 8, 9, 10, 11, 12, 13, 14

[14] Jihyong Oh and Munchurl Kim. Demfi: deep joint deblurring and multi-frame interpolation with flow-guided attentive correlation and recursive boosting. In *European Conference on Computer Vision*, pages 198–215. Springer, 2022. 2

[15] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European Conference on Computer Vision*, pages 327–343. Springer, 2020. 1

[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 3

[17] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 4

[18] 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, 8, 9, 10, 11, 12, 13, 14

[19] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 6, 8, 9, 10, 11, 12, 13, 14

[20] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4472–4480, 2017. 1

[21] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018. 1, 2, 6, 8, 9, 10, 11, 12, 13, 14

[22] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[23] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for

Extended Figure S.13: Supplemental qualitative results for generalization evaluation.

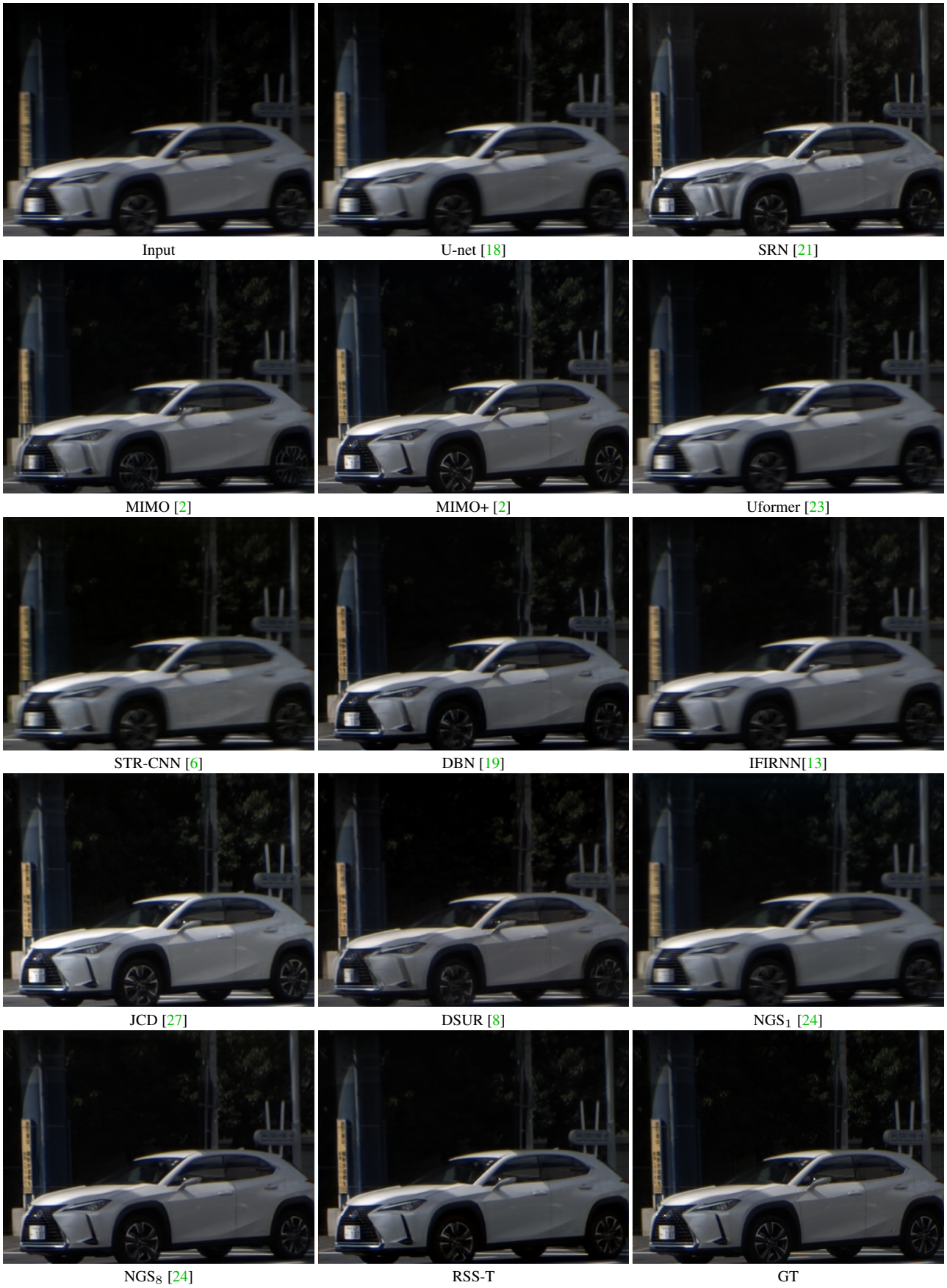
image restoration. *arXiv preprint arXiv:2106.03106*, 2021. 1, 8, 9, 10, 11, 12, 13, 14

[24] Zhixiang Wang, Xiang Ji, Jia-Bin Huang, Shin'ichi Satoh, Xiao Zhou, and Yinqiang Zheng. Neural global shutter: Learn to restore video from a rolling shutter camera with global reset feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17794–17803, 2022. 2, 5, 6, 8, 9, 10, 11, 12, 13, 14

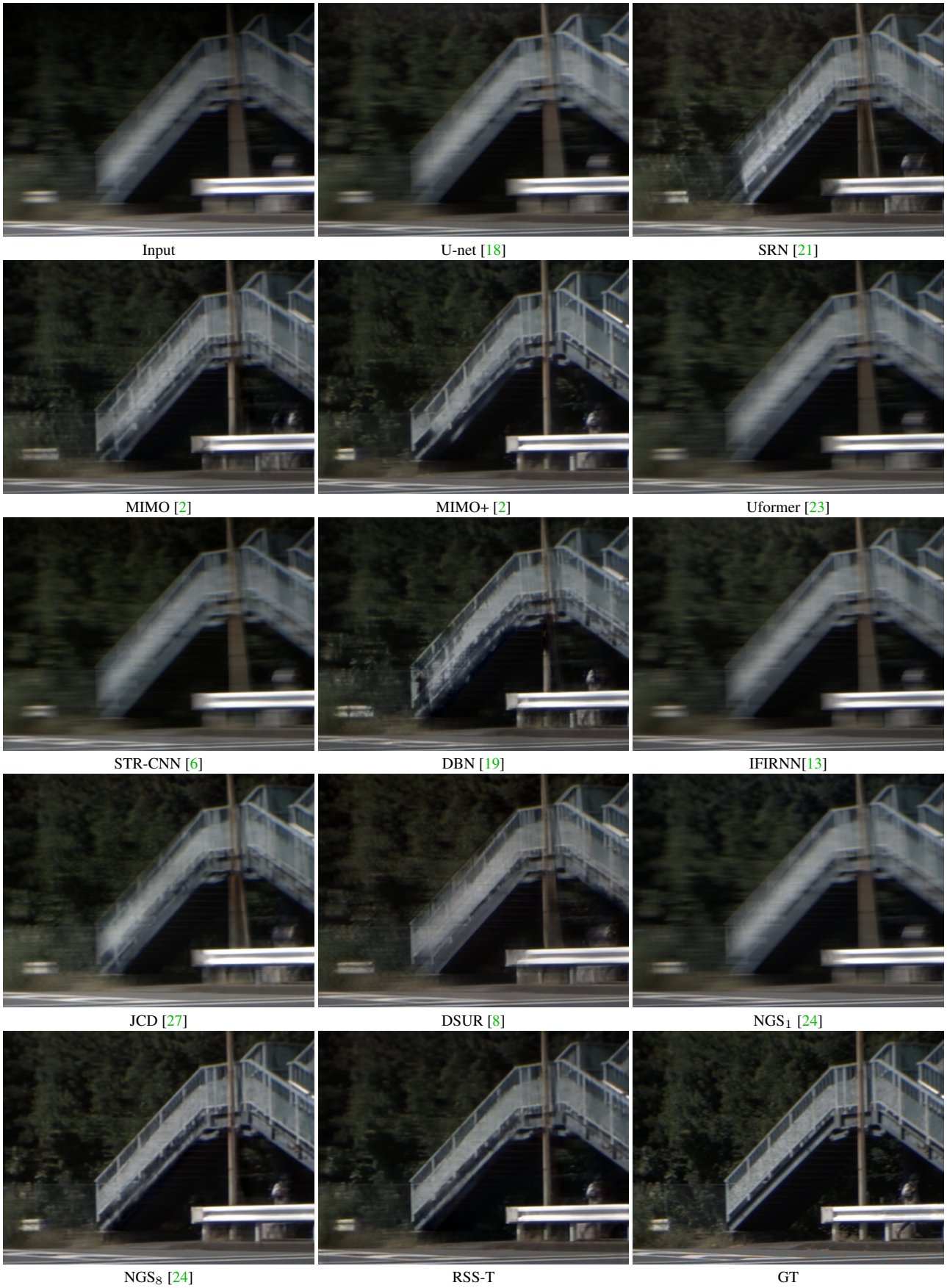
[25] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *European Conference on Computer Vision*, pages 492–511. Springer, 2020. 2

[26] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 2, 6

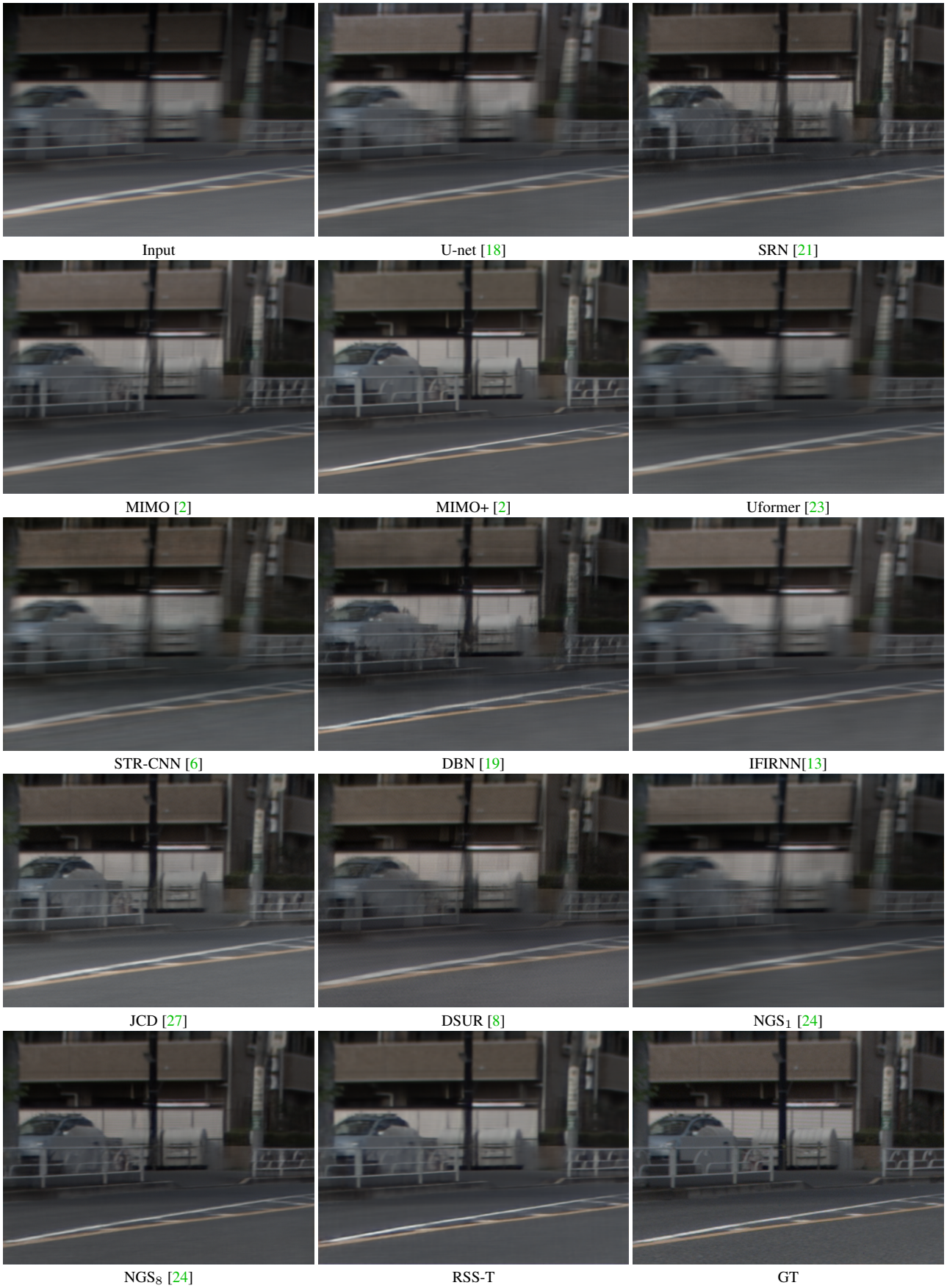
[27] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9219–9228, 2021. 6, 8, 9, 10, 11, 12, 13, 14



Extended Figure S.14: Additional qualitative results.

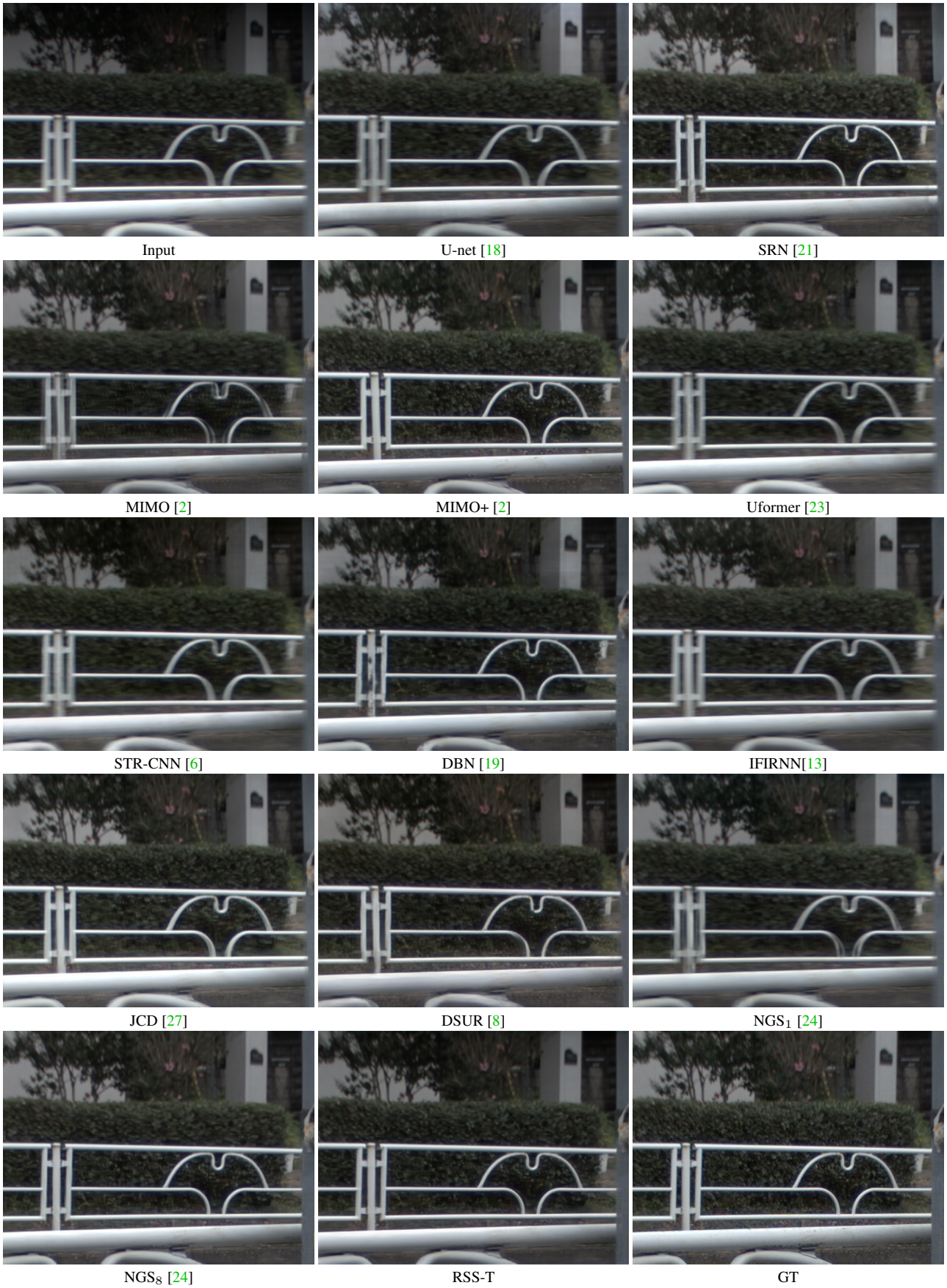


Extended Figure S.15: Additional qualitative results.



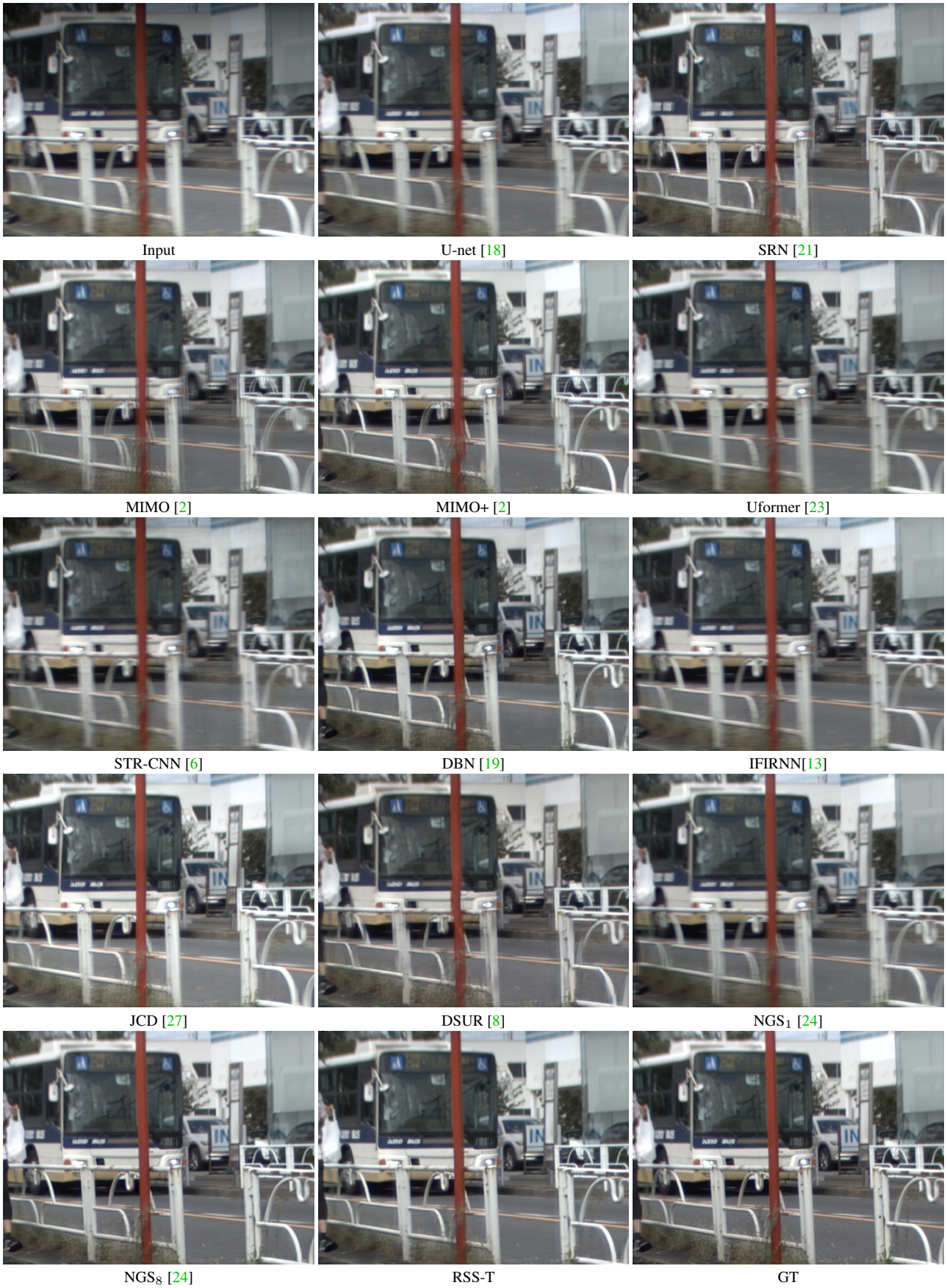
Extended Figure S.16: Additional qualitative results.

Extended Figure S.17: Additional qualitative results.



Extended Figure S.18: Additional qualitative results.

Extended Figure S.19: Additional qualitative results.



Extended Figure S.20: Additional qualitative results.