Enriched CNN-Transformer Feature Aggregation Networks for Super-Resolution –Supplementary Material–

Jinsu Yoo[1] Taehoon Kim[2] Sihaeng Lee[2] Seung Hwan Kim[2] Honglak Lee[2] Tae Hyun Kim[1]

[1]Hanyang University [2]LG AI Research

In this supplementary material, we provide additional details, experimental settings, and results of our proposed method. First, we describe the details of the proposed Fusion Block and CSTA. Next, we report more results related to the model size analysis in the main manuscript. Then, we show the applicability of our approach to another image restoration task (*i.e.*, JPEG artifact removal). Finally, we provide more visual results of feature visualization and qualitative comparisons.

1. Fusion Block

Settings for the ablation in the main manuscript: We elaborate on detailed settings of our fusion strategies in the main manuscript. For the ablation experiment without intermediate fusion (third column in Table 2 in the main paper), we concatenate output features $\mathbf{T}_4$ and $\mathbf{F}_4$ from separated branches and pass them to the tail module. For the experiments with the unidirectional flow (fourth and fifth columns in Table 2 in the main paper), we do not split feature $\mathbf{M}_i$ into two. Instead, we reduce channel dimension from $2c$ to c with a single convolution layer and transfer it to branch chosen to receive fused representation (see Figure 1).

Experiment on lateral connection: Since the dimension of the feature from the CNN branch and image-likely rearranged feature from the transformer branch are the same (*i.e.*, $c \times h \times w$), one can laterally connect two features with element-wise summation rather than concatenation in Equation 7 in the main manuscript. We compare the performance of different lateral connection choices in Table 1 and observe that concatenation gives slightly better results than summation. Therefore, we finalize our lateral connection to concatenating two features.

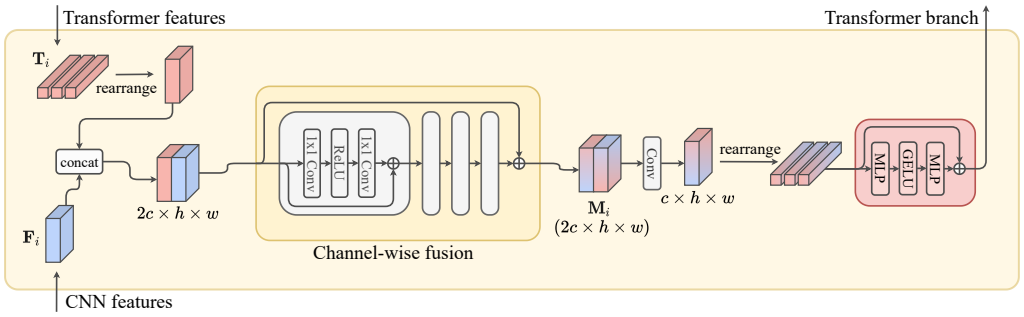
Method	Lateral connection	Set14/Urban100
ACT (Ours)	element-wise summation	34.57/34.05
ACT (Ours)	concatenation	**34.60/34.07**

Table 1: Comparison on different lateral connections of two branches in our Fusion Block, reported in PSNR value.

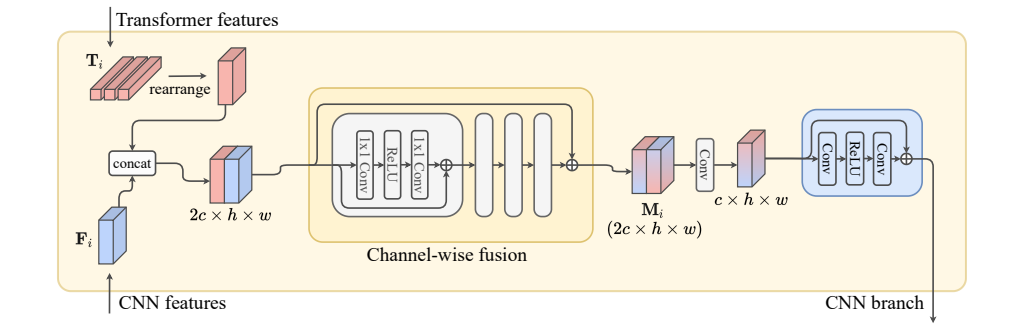
2. Cross-Scale Token Attention

Pseudocode of CSTA: Algorithm 1 provides pseudocode of CSTA for better understanding. In particular, we split the hidden dimension of $\mathbf{T}$ into two and directly utilize $\mathbf{T}^a$ for $\mathbf{T}^s$ while reformulating $\mathbf{T}^b$ to acquire $\mathbf{T}^l$ as elaborated in the main manuscript. It is worth noting that such rearrangement is similar to soft-split, which is introduced in T2T-ViT [13] in that tokens are re-structured by overlapping. Unlike soft-split, we rearrange only a part of the hidden dimension within token $\mathbf{T}$ with a larger token size to perform cross-attention efficiently. Moreover, our rearrangement aims to acquire numerous larger tokens (patches) to exploit recurring patches across different scales within the input image.

Settings for the ablation in the main manuscript: We elaborate details about *Impact of more token scales* experiment in our ablation on CSTA in the main manuscript. Specifically, we demonstrate how we perform cross-attention by introducing three different token scales (3rd row in Table 3c in the main paper). To enable our network to leverage patch-recurrence across various scales while maintaining similar computational cost, we first split input token $\mathbf{T} \in \mathbb{R}^{n \times d}$ into four tokens $\{\mathbf{T}_i\}_{i=1}^{4} \in \mathbb{R}^{n \times d/4}$ before cross-attention. Then, we rearrange $\mathbf{T}_2$ and $\mathbf{T}_4$ with token size of 6×6 and 12×12, respectively, with the same strides of 3, while $\mathbf{T}_1$ and $\mathbf{T}_3$ keep their token size of 3×3. We perform two independent cross-attention using ($\mathbf{T}_1$, $\mathbf{T}_2$) and ($\mathbf{T}_3$, $\mathbf{T}_4$) pair as elaborated in the main manuscript. Consequently, the network can utilize multi-scale information across various scales. Lastly, we re-concatenate four tokens into one after rearranging token $\mathbf{T}_2$ and $\mathbf{T}_4$ to include a token size of 3×3.



(a) Fusion Block (unidirectional, CNN to transformer branch)



(b) Fusion Block (unidirectional, Transformer to CNN branch)

Figure 1: Illustration of unidirectional Fusion Blocks, used for the main manuscript's ablation experiments.

IPT [1]	**SwinIR** [4]	**ACT** (Ours)
0.586s	0.528s	0.566s

Table 2: Runtime comparison.

3. Model Size Analysis

For model size analysis in Table 5 in the main manuscript, we count FLOPs for 48×48 image following IPT [1] using the open source. [1] Moreover, we compare the runtime of our model with recently proposed transformer-based SR methods [1, 4]. We average five runs on the same setting as in Table 5 in the main paper with Ryzen 2950X CPU and NVIDIA 2080 Ti. The result in Table 2 shows that ours is competitive in runtime.

4. Applicability to other Image Restoration Tasks

We investigate the applicability of our network to a challenging restoration task; color JPEG artifact removal. To do so, we only modify the tail part of the network to include a single convolutional layer while discarding PixelShuffle upsampler [11]. We train our network with the same training configurations for SR, and the training patch size is 48×48. We compare our ACT with the state-of-the-art color JPEG artifact removal methods, including QGAC [2] and FBCNN-C [3] for quality factors of 10, 20, 30, and 40. We evaluate performance on LIVE1 [10], BSDS500 [6], and ICB [9] datasets with three different metrics including PSNR, SSIM, and PSNR-B values following the baselines [2, 3]. We report the quantitative comparison in Table 3. As shown in the table, our ACT shows promising results despite lacking task-specific architectural design. Moreover, we visually compare our method against the baselines in Figure 2. Compared to the baselines, our ACT accurately restores corrupted images, including natural scenes, sharp textures, and characters. This result implies the possibility of applying our ACT to various restoration tasks similar to the recent image restoration transformers [1, 4, 12].

[1] https://github.com/facebookresearch/fvcore

5. Feature Visualization

In Figure 3, we provide more feature visualizations to understand the role of two branches.

6. Additional Qualitative Results

To further demonstrate the superiority of our proposed method, we provide more visual comparisons with six state-of-the-art SR methods: EDSR [5], RCAN [14], HAN [8], NLSA [7], IPT [1], and SwinIR [4]. The visual comparisons are shown in Figure 4 and Figure 5.

Dataset	QF	JPEG	QGAC [2]	FBCNN-C [3]	ACT (Ours)
LIVE1 [10]	10	25.69/0.743/24.20	27.62/0.804/27.43	27.77/0.803/27.51	**27.94/0.808/27.63**
	20	28.06/0.826/26.49	29.88/0.868/29.56	30.11/0.868/29.70	**30.39/0.874/29.96**
	30	29.37/0.861/27.84	31.17/0.896/30.77	31.43/0.897/30.92	**31.77/0.902/31.26**
	40	30.28/0.882/28.84	32.05/0.912/31.61	32.34/0.913/31.80	**32.70/0.917/32.16**
BSDS500 [6]	10	25.84/0.741/24.13	27.74/**0.802**/27.47	**27.85**/0.799/**27.52**	27.84/0.800/27.46
	20	28.21/0.827/26.37	30.01/**0.869**/29.53	30.14/0.867/29.56	**30.20**/0.868/**29.58**
	30	29.57/0.865/27.72	31.33/**0.898**/30.70	31.45/0.897/30.72	**31.51**/0.897/**30.77**
	40	30.52/0.887/28.69	32.25/**0.915**/31.50	32.36/0.913/31.52	**32.41**/0.914/**31.56**
ICB [9]	10	29.44/0.757/28.53	32.06/**0.816**/32.04	32.18/0.815/32.15	**32.20/0.816/32.17**
	20	32.01/0.806/31.11	34.13/0.843/34.10	34.38/**0.844**/34.34	**34.50/0.844/34.46**
	30	33.20/0.831/32.35	35.07/0.857/35.02	35.41/0.857/35.35	**35.61/0.859/35.55**
	40	33.95/0.840/33.14	32.25/**0.915**/31.50	36.02/0.866/35.95	**36.21**/0.868/**36.13**

Table 3: PSNR/SSIM/PSNRB comparison of different state-of-the-art methods on color JPEG artifact removal. Our ACT shows competitive performance over baselines. Performances for the baselines are borrowed from [3]. The best and the second-best values are highlighted with **bold** and underline, respectively.

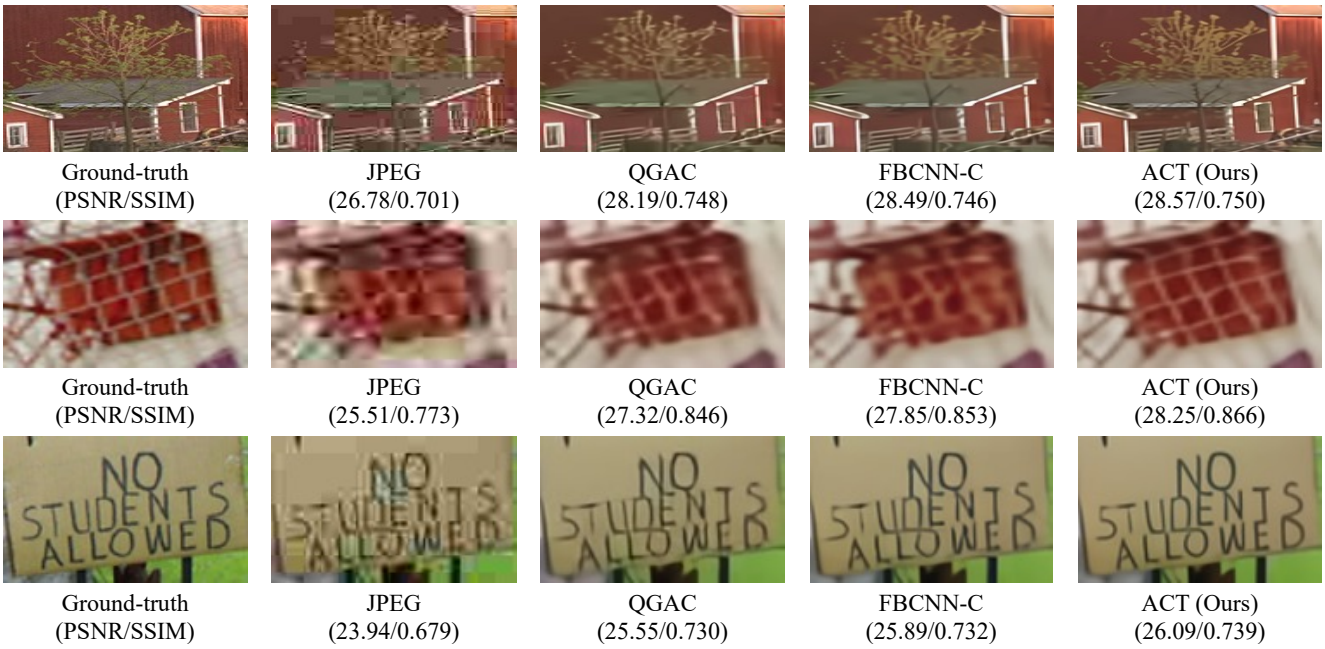


Figure 2: Visual comparison of our ACT against state-of-the-art color JPEG artifact removal methods [2, 3] with quality factor of 10. Our ACT better removes the artifacts and produces accurate structures than the baselines.

References

[1] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[2] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Quantization guided jpeg artifact correction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[3] Jiaxi Jiang, Kai Zhang, and Radu Timofte. Towards flexible blind jpeg artifacts removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[4] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021.

[5] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

Algorithm 1: Pseudocode of CSTA in a PyTorch-like style.

```
# b, n, d:  batch size, number of tokens, and hidden dimension of T
# n', d':  number of tokens and hidden dimension of T^l directly after acquiring
 large tokens from T^b by rearrangement
# h, w:  height and width of patches

import torch
import torch.nn.functional as F

def CSTA(T):
   # split T (b × n × d) into T^a and T^b
   T_a, T_b = torch.split(T, d//2, dim=2)
   # acquire T^s from T^a
   T_s = T_a
   # acquire T^l from T^b by rearrangement
   T_l = F.fold(T_b, output_size=(h, w), kernel_size=token_size,
    stride=token_size)
   T_l = F.unfold(T_l, kernel_size=token_size*2, stride=token_size)

   # project tokens into query, key, and value
   T_l = mlp_blk_before_attn(T_l) # reduce dimension from d' to d/2
   q_l, k_l, v_l = project(T_l)
   q_s, k_s, v_s = project(T_s)

   # perform cross attention
   T_l = attention(query=q_l, key=k_s, value=v_s)
   T_s = attention(query=q_s, key=k_l, value=v_l)

   # increase hidden dimension of T^l from d/2 to d'
   T_l = mlp_blk_after_attn(T_l)

   # rearrange T^l from (b × n' × d') to (b × n × (d/2))
   T_l = F.fold(T_l, output_size=(h, w), kernel_size=token_size*2,
    stride=token_size)
   T_l = F.unfold(T_l, kernel_size=token_size, stride=token_size)

   # concatenate T^s and T^l into T (b × n × d)
   T = torch.cat((T_s, T_l), dim=2)

   return T
```


[6] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001.

[7] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[8] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[9] Rawzor. Image compression benchmark.

[10] HR Sheikh. Live image quality assessment database release 2. *http://live.ece.utexas.edu/research/quality*, 2005.

[11] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ground-truth Transformer branch CNN branch

Figure 3: Feature map visualizations of transformer branch and CNN branch. Brighter color indicates higher value.

[12] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021.

[13] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[14] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

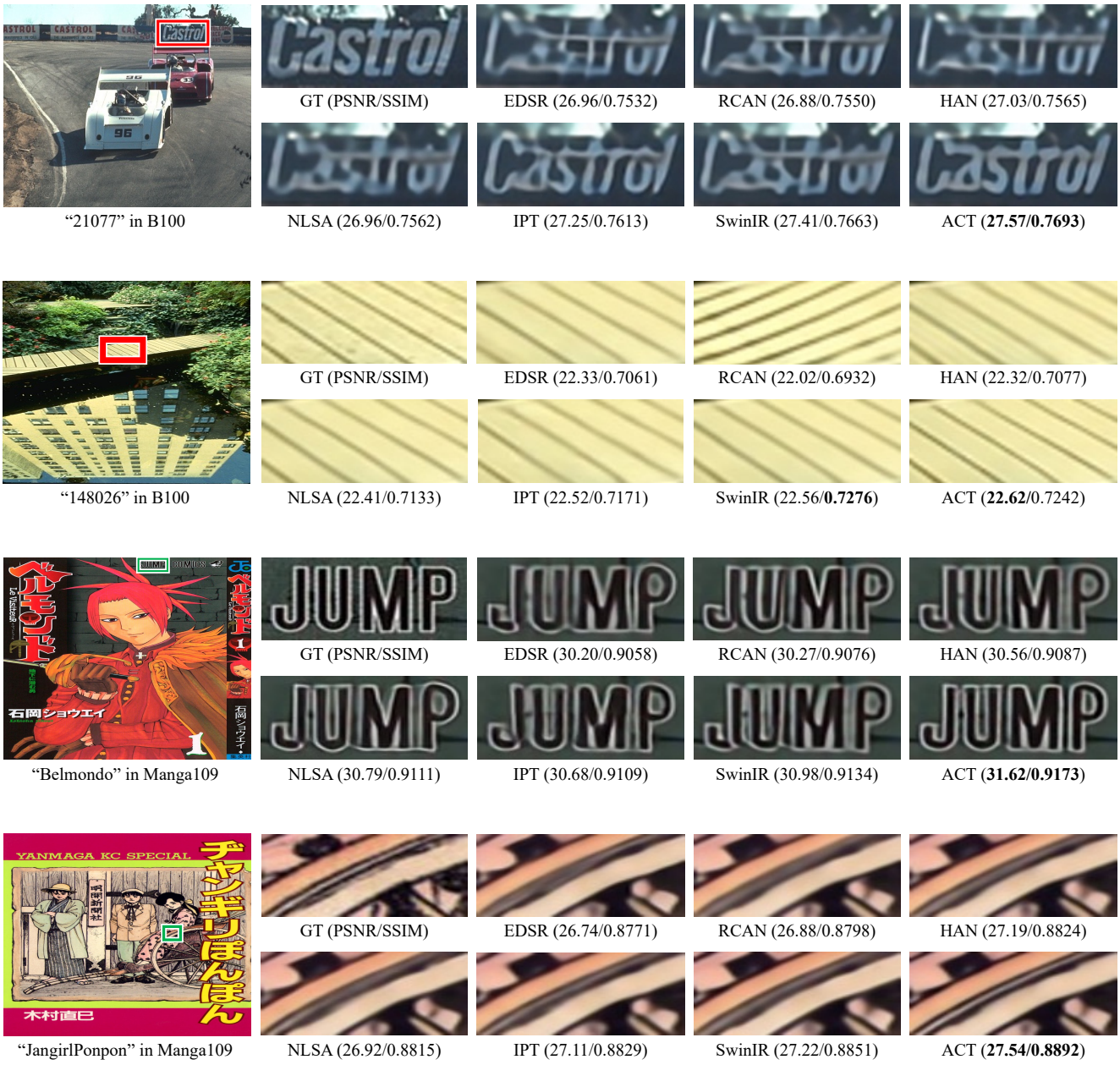


Figure 4: Visual comparison of the proposed method against various state-of-the-art methods for $\times 4$ SR.

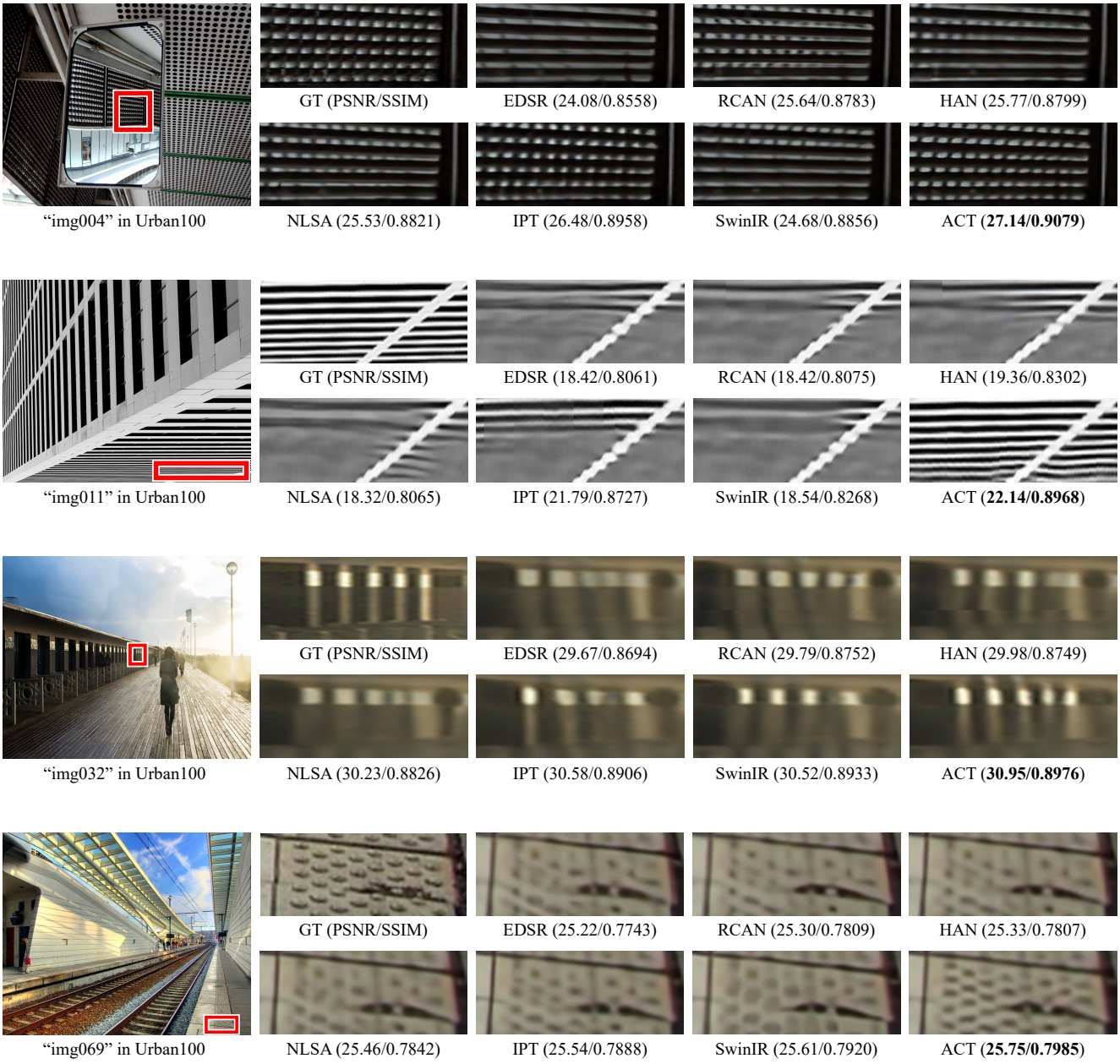


Figure 5: Visual comparison of the proposed method against various state-of-the-art methods for $\times 4$ SR.