Supplemental Material:
Exploring Sparsity in Image Super-Resolution for Efficient Inference

Longguang Wang[1], Xiaoyu Dong[2,3], Yingqian Wang[1], Xinyi Ying[1], Zaiping Lin[1], Wei An[1], Yulan Guo[1*]
[1]National University of Defense Technology [2]The University of Tokyo [3]RIKEN AIP
{wanglongguang15,yulan.guo}@nudt.edu.cn

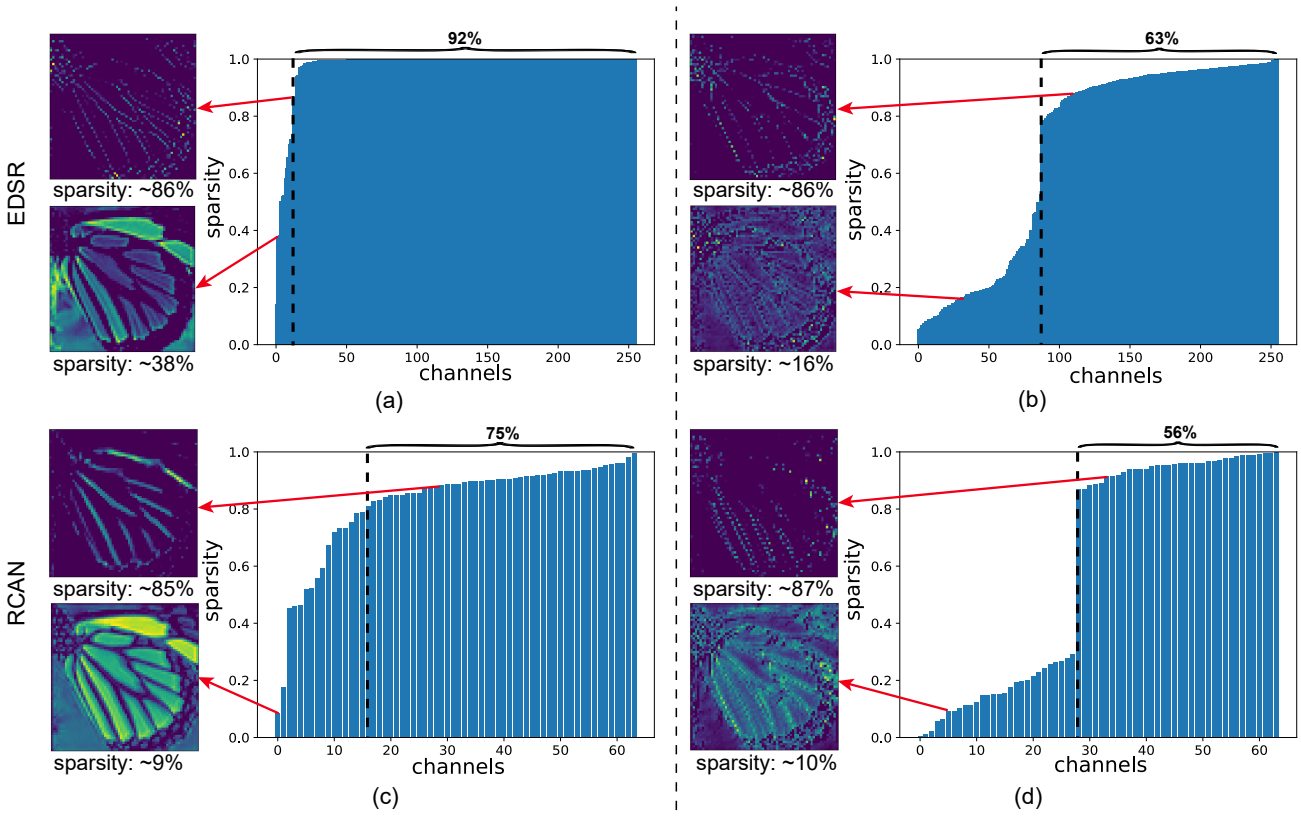

Figure I: Visualization of feature maps after the ReLU layers in different backbone blocks of EDSR and RCAN. (a) and (c) are from the first backbone block, while (b) and (d) are from the last backbone block.

Section I presents more examples of feature visualization for the analyses in Section 3. Section II provides additional analyses regarding our sparse masks. Section III investigates the compatibility of our sparse masks with other SR networks. Finally, Section IV includes additional visual results on different datasets.

I. Feature Sparsity in SR Networks

Figure I shows the feature maps after the ReLU layers in the backbone blocks of EDSR and RCAN. It can be observed that a considerable number of channels are quite sparse (sparsity ≥ 0.8), with only "important" regions (*e.g.*, edge and texture regions) being activated. For those "unimportant" regions (*e.g.*, flat regions), only a few channels are activated in EDSR and RCAN. We also illustrate the average feature sparsity achieved by RCAN on B100 in Fig. II. Specifically, the feature sparsities of corresponding channels are averaged over 100 images in B100. It can be observed that the results are consistent with Fig. I, which further demonstrates the feature sparsity in SR networks.

Figure III illustrates the feature maps in our SMSR. As

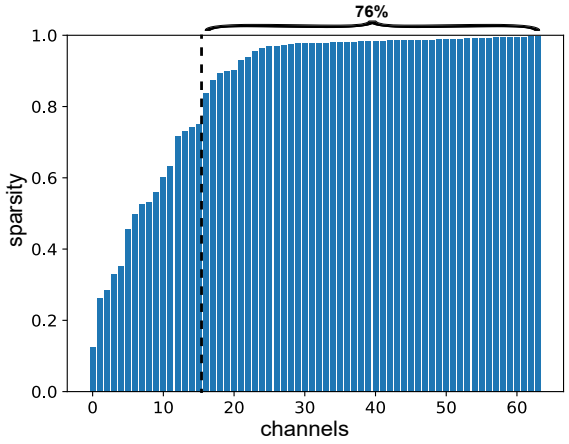

Figure II: Feature sparsity averaged over B100.

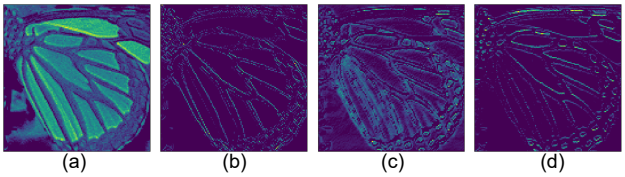

Figure III: Visualization of feature maps in our SMSR. (a) and (b) are from the first SMM, while (c) and (d) are from the last SMM. (a) and (c) are "dense" features marked by M^{ch}, while (b) and (d) are "sparse" features marked by $(1-M^{ch})$.

we can see, feature maps marked by M^{ch} preserve beneficial information in flat regions. In contrast, only regions of edges and textures are activated in the feature maps marked by $(1-M^{ch})$.

II. Additional Analyses of Sparse Masks

Effectiveness of Channel Masks. In addition to spatial masks, channel masks work in an orthogonal dimension to enable our network to localize redundant computation at a fine-grained level. Without channel masks, out network suffers a conflict between efficiency (reducing redundant computation) and performance (preserving necessary computation) since redundant computation in channel dimension cannot be well handled. From Table I we can see that our SMSR achieves higher sparsity (*i.e.*, lower computational cost) with improved performance (38.00/33.64 vs. 37.97/33.60) if channel masks are used. This clearly demonstrates the effectiveness of channel masks for accurate localization of redundant computation.

Static Channel Masks vs. Dynamic Channel Masks. In our SMSR, static channel masks are used to mark redundant channels since we observe that the indices of channels with "dense" and "sparse" feature maps are almost consistent in state-of-the-art SR networks for different inputs, as shown in Fig. IV. That is, the redundancy in channel dimension for "unimportant" regions (*i.e.*, flat regions) has little relation to the input. To further demonstrate this, we introduce a network variant that predicts dynamic channel masks according to the input image at inference time. The comparative results are presented in Table I. It can be observed that dynamic channel masks do not introduce performance gain but include additional parameters and computational cost. Therefore, static channel masks are used in our SMSR.

Table I: Comparative results achieved on Set5 and Set14 for $\times 2$ SR.

Model	M^{ch}	#Params	Set5			Set14		
			Sparsity	PSNR	SSIM	Sparsity	PSNR	SSIM
SMSR	✗	985K	0.51	37.97	0.9600	0.42	33.60	0.9176
SMSR	Dynamic	1012K	0.56	37.98	0.9602	0.46	33.62	0.9178
SMSR	Static	985K	0.58	38.00	0.9601	0.46	33.64	0.9179

Visualization of Sparse Masks. Figure V visualizes sparse masks generated within different SMMs. From SMM-1 to SMM-3, M^{spa} learns to mark more locations as "important" ones while M^{ch} reduces channels preserved for those "unimportant" locations (*i.e.*, blue regions in M^{ch}). Moreover, SMM-4 and SMM-5 mainly focus on refining the features on a few locations only.

We further investigate the sparsities achieved by our SMMs for different scale factors. Specifically, we feed an LR image ($\times 2$ downsampled) to $\times 2/3/4$ SMSR networks and compare the sparsities in their SMMs. As shown in Fig VI, the sparsities decrease for larger scale factors in most SMMs. Since more details need to be reconstructed for larger scale factors, more locations are marked as "important" ones (with sparities being decreased).

Learning-based Masks vs. Gradient-based Masks. We compare learning-based masks with gradient-based masks in Fig. VII. Compared to learning-based masks, gradient-based spatial masks are fixed throughout the network and have limited flexibility. Specifically, gradient-based masks have to activate more channels with "dense" features (*i.e.*, the blue regions in M^{ch}) to preserve sufficient information in those regions uncovered by the spatial masks. Therefore, it is difficult for gradient-based masks to obtain fine-grained localization of redundant computation. In contrast, our learning-based masks can accurately localize redundant computation to facilitate our SMSR to achieve better performance, as demonstrated in Section 5.2.

III. Compatibility with Other SR Networks

We conduct experiments by applying our sparse masks to existing SR networks to show their compatibility with other architectures. Specifically, we use SRResNet [2] and IMDN [1] as the baselines and compare our sparse masks with recent network compression techniques [3, 4]. SRResNet is a widely-used baseline while IMDN is a highly-optimized efficient SR network. Quantitative results are presented in Table II.

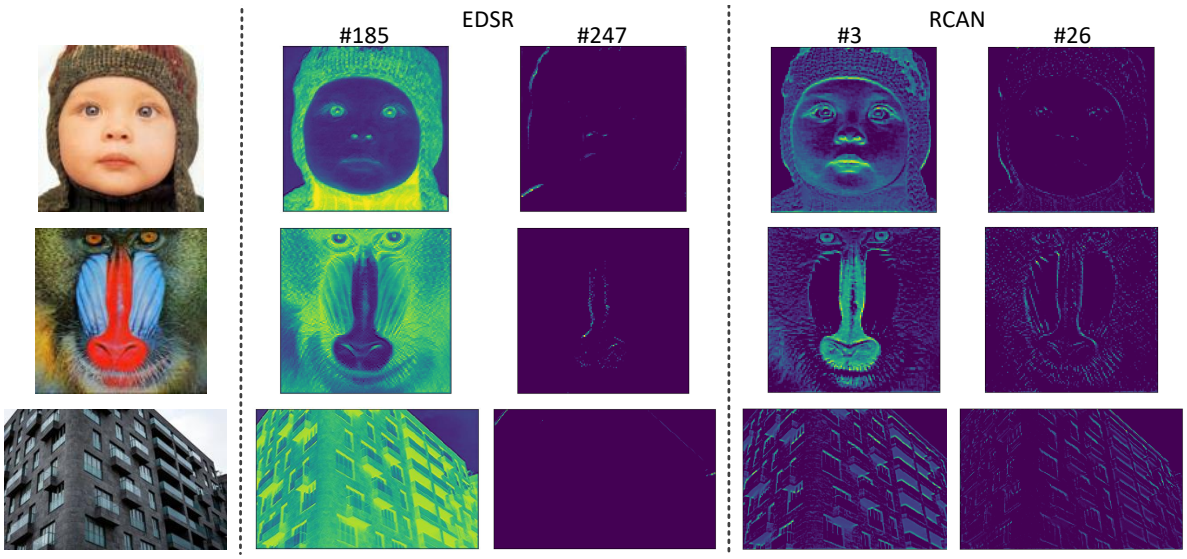

Figure IV: Visualization of feature maps in the first backbone block of EDSR and RCAN. For different input images, channel #185 in EDSR and channel #3 in RCAN consistently carry "dense" feature maps, while channel #247 in EDSR and channel #26 in RCAN carry "sparse" feature maps.

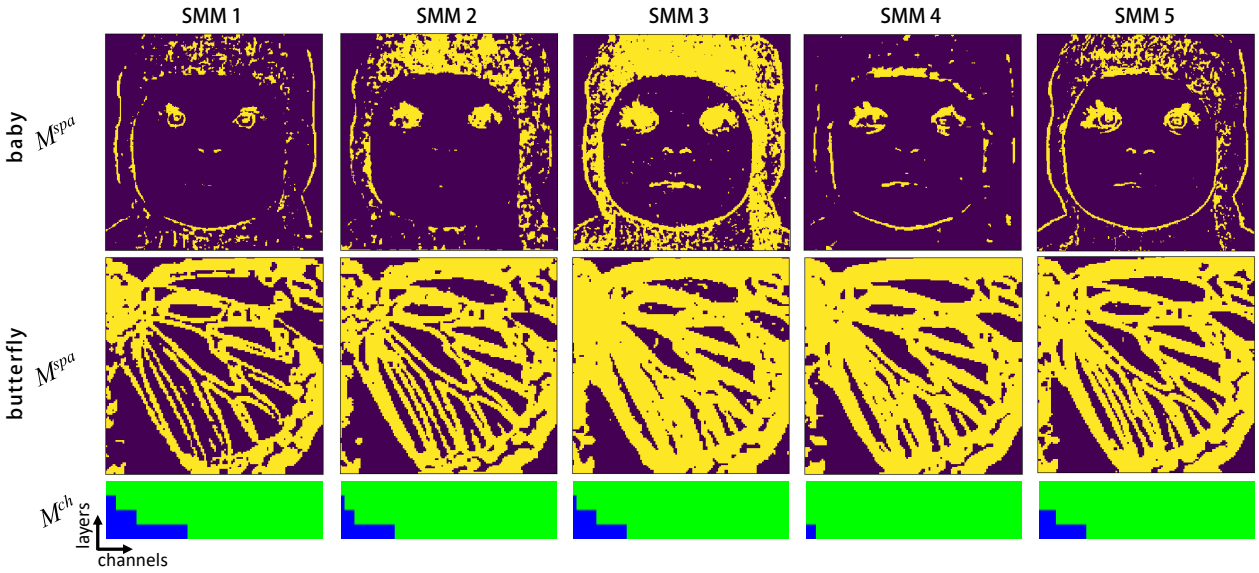

Figure V: Visualization of sparse masks in different SMMs on *baby* and *butterfly* for $\times 2$ SR.

Compared to other network compression techniques, our sparse masks produce higher PSNR results with comparable computational complexity in terms of FLOPs. Since our sparse masks consider redundancy in both spatial and channel dimensions to localize redundant computation at a fine-grained level, superior performance can be achieved. Moreover, our sparse masks are also compatible with well-designed IMDN to further reduce its computational cost while maintaining comparable performance. This clearly demonstrates the good compatibility and effectiveness of our sparse masks.

IV. Additional Visual Results

Figure VIII provides additional visual results achieved on three images from the Urban100, Set14 and Manga109 datasets. It can be observed from the zoom-in regions that our SMSR recovers finer details with better perceptual quality while other methods suffer obvious blurring or distorted artifacts.

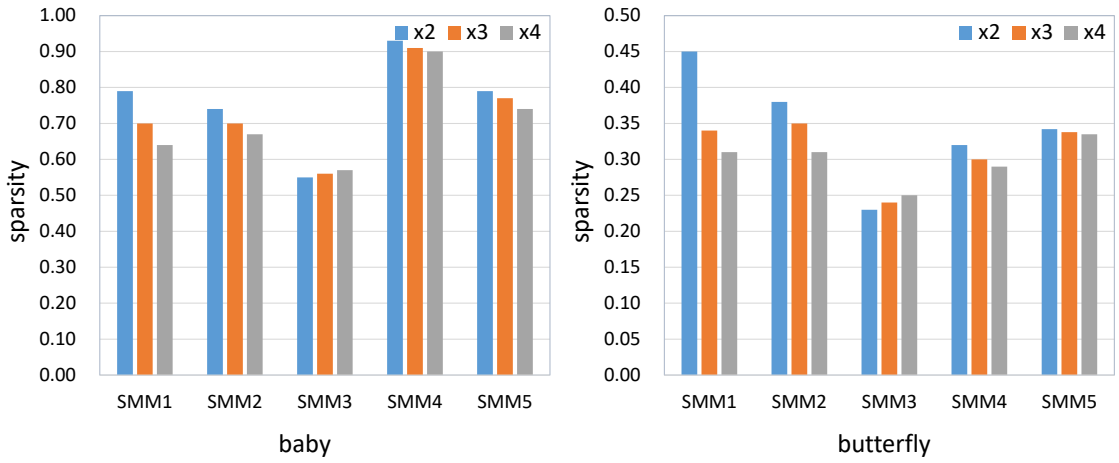

Figure VI: Comparison of sparsities achieved in different SMMs on *baby* and *butterfly* for different scale factors.

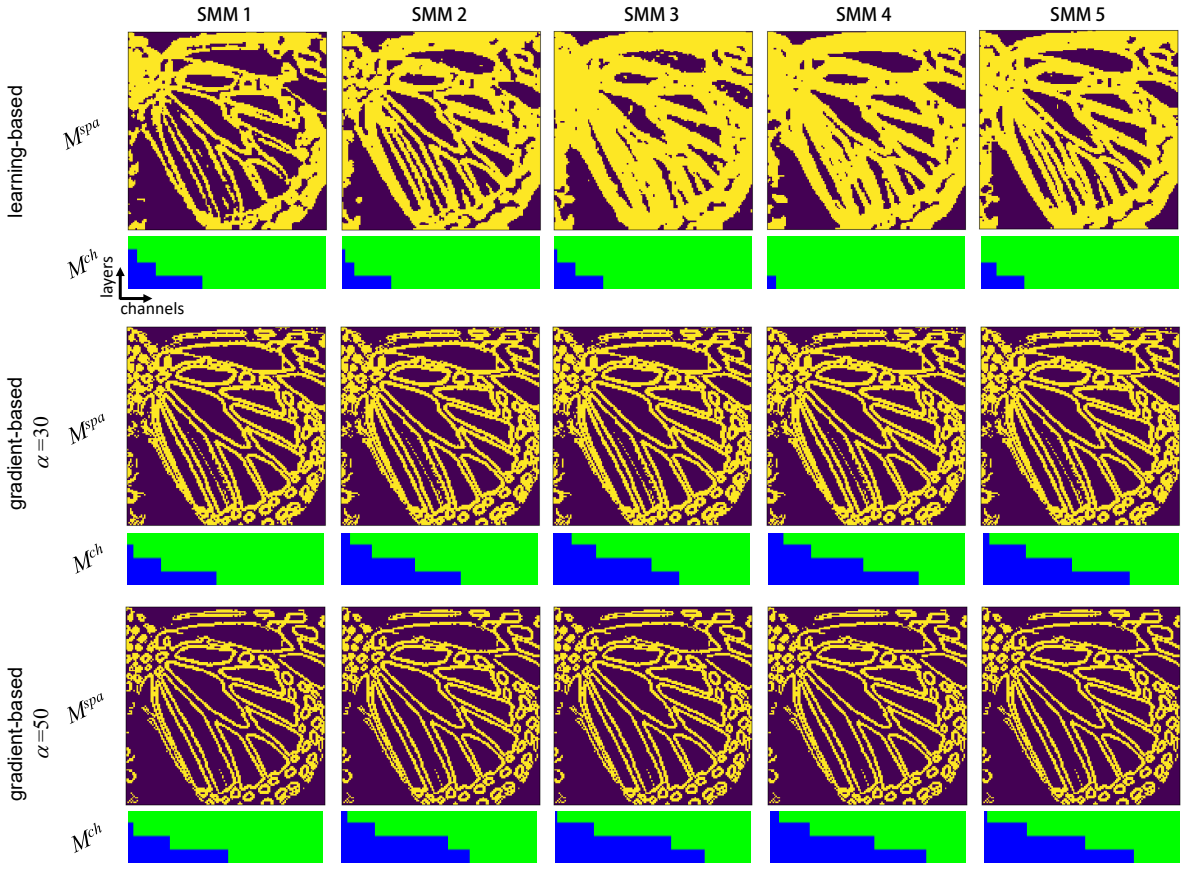

Figure VII: Comparison of learning-based and gradient-based masks.

Acknowledge

The authors would like to thank anonymous reviewers for their insightful suggestions. Xiaoyu Dong was supported by RIKEN Junior Research Associate Program.

References

[1] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*. ACM Press, 2019. 2

Table II: PSNR results achieved for $\times 4$ SR. FLOPs is computed based on HR images with a resolution of 720p (1280×720).

Model	Method	#Params.	FLOPs	Set5	Set14	B100	Urban100
SRResNet	Baseline	1.54M	112.11G	32.03	28.50	27.52	25.88
	DHP [4]	0.95M	69.29G($\downarrow$ 38.2%)	31.97	28.47	27.48	25.76
	Basis [3]	0.74M	67.51G($\downarrow$ 39.8%)	31.90	28.42	27.44	25.65
	Ours	1.58M	72.65G($\downarrow$ 35.2%)	**32.03**	**28.52**	**27.52**	**25.97**
	DHP [4]	0.64M	46.82G($\downarrow$ 58.2%)	31.90	28.45	27.47	25.72
	Basis [3]	0.60M	55.72G($\downarrow$ 50.3%)	31.84	28.38	27.39	25.54
	Ours	1.54M	54.26G($\downarrow$ 51.6%)	**31.95**	**28.47**	**27.49**	**25.89**
IMDN	Baseline	0.72M	41.22G	32.21	28.58	27.56	26.04
	Basis [3]	0.60M	34.16G ($\downarrow$ 17.1%)	32.04	28.47	27.49	25.79
	Ours	0.77M	32.12G ($\downarrow$ 22.1%)	**32.11**	**28.55**	**27.53**	**26.02**
	Basis [3]	0.45M	24.52G ($\downarrow$ 40.5%)	31.98	28.44	27.46	25.74
	Ours	0.77M	23.77G ($\downarrow$ 42.3%)	**32.08**	**28.52**	**27.51**	**25.98**

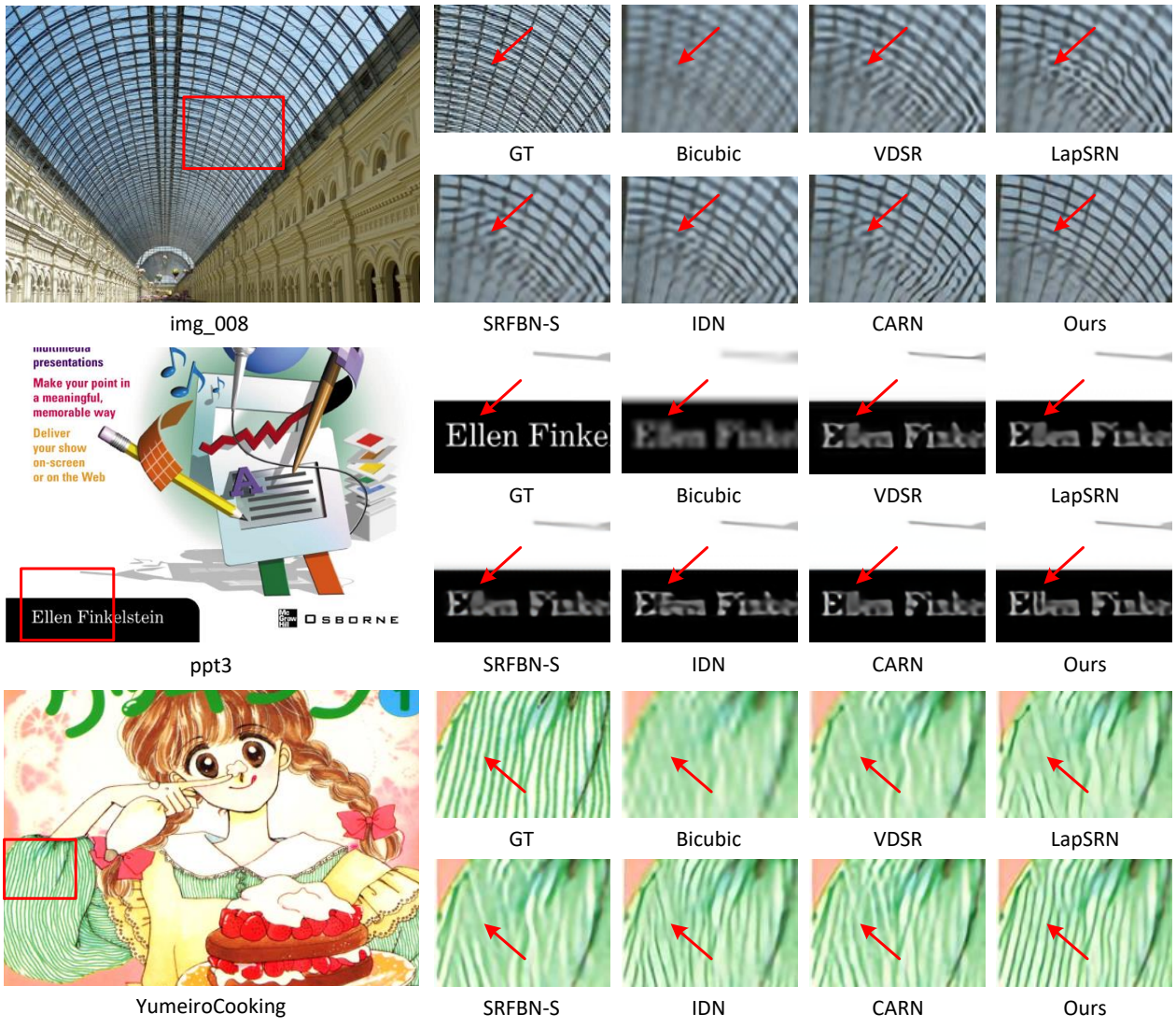

Figure VIII: Visual comparison on Urban100, Set14, and Manga109 for $\times 4$ SR.

[2] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 105–114, 2017. 2

[3] Yawei Li, Shuhang Gu, Luc Van Gool, and Radu Timofte. Learning filter basis for convolutional neural network compression. In *ICCV*, pages 5623–5632, 2019. 2, 5

[4] Yawei Li, Shuhang Gu, Kai Zhang, Luc Van Gool, and Radu Timofte. Dhp: Differentiable meta pruning via hypernetworks. In *ECCV*, 2020. 2, 5