Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution Supplementary Material

Wei-Sheng Lai[1] Jia-Bin Huang[2] Narendra Ahuja[3] Ming-Hsuan Yang[1]
[1]University of California, Merced [2]Virginia Tech [3]University of Illinois, Urbana-Champaign
http://vllab.ucmerced.edu/wlai24/LapSRN

1. Overview

In this supplementary document, we present additional results to complement the paper. First, we provide the detailed configurations and parameters of the proposed LapSRN. Second, we show the comparisons with the network architecture of LAPGAN [3] and the adversarial loss training. We also provide analysis on different training datasets used in existing methods. Third, we present the run time evaluation and more qualitative comparisons with the state-of-the-art algorithms.

2. Network Architectures

The proposed LapSRN consists of two branches:

- *Feature extraction branch* uses a cascade of convolutional neural networks to predict residual images at multiple scales.
- *Image reconstruction branch* reconstructs high-resolution (HR) images in a coarse-to-fine manner.

Table 1 lists the detailed configurations of our 8× model with depth $d = 5$ at each level. All the convolutional layers have a stride of 1 and pad the feature maps with 1 pixel to keep the size of the feature maps unchanged at each level. All the transposed convolutional layers upscale feature maps by 2 and crop the boundaries of feature maps by 1 pixel.

Table 1: Detailed configuration of the proposed LapSRN for the 8× scale factor. "conv" and "convt" denote the convolution layers and transposed convolution layers, respectively.

Feature Extraction Branch				**Image Reconstruction Branch**			
Input	Output	Kernel size	Output size	Input	Output	Kernel size	Output size
LR	conv-0	$3 \times 3 \times 1 \times 64$	$16 \times 16 \times 64$	LR	convt-I1	$4 \times 4 \times 1 \times 1$	$32 \times 32 \times 1$
conv-0	conv1-1	$3 \times 3 \times 64 \times 64$	$16 \times 16 \times 64$	convt-F1	conv-R1	$3 \times 3 \times 64 \times 1$	$32 \times 32 \times 1$
conv1-1	conv1-2	$3 \times 3 \times 64 \times 64$	$16 \times 16 \times 64$	conv-I1 & conv-R1	HR-2×	element-wise sum	$32 \times 32 \times 1$
conv1-2	conv1-3	$3 \times 3 \times 64 \times 64$	$16 \times 16 \times 64$				
conv1-3	conv1-4	$3 \times 3 \times 64 \times 64$	$16 \times 16 \times 64$				
conv1-4	conv1-5	$3 \times 3 \times 64 \times 64$	$16 \times 16 \times 64$				
conv1-5	convt-F1	$4 \times 4 \times 64 \times 64$	$32 \times 32 \times 64$				
convt-F1	conv2-1	$3 \times 3 \times 64 \times 64$	$32 \times 32 \times 64$	HR-2×	convt-I2	$4 \times 4 \times 1 \times 1$	$64 \times 64 \times 1$
conv2-1	conv2-2	$3 \times 3 \times 64 \times 64$	$32 \times 32 \times 64$	convt-F2	conv-R2	$3 \times 3 \times 64 \times 1$	$64 \times 64 \times 1$
conv2-2	conv2-3	$3 \times 3 \times 64 \times 64$	$32 \times 32 \times 64$	conv-I2 & conv-R2	HR-4×	element-wise sum	$64 \times 64 \times 1$
conv2-3	conv2-4	$3 \times 3 \times 64 \times 64$	$32 \times 32 \times 64$				
conv2-4	conv2-5	$3 \times 3 \times 64 \times 64$	$32 \times 32 \times 64$				
conv2-5	convt-F2	$4 \times 4 \times 64 \times 64$	$64 \times 64 \times 64$				
convt-F2	conv3-1	$3 \times 3 \times 64 \times 64$	$64 \times 64 \times 64$	HR-4×	convt-I3	$4 \times 4 \times 1 \times 1$	$128 \times 128 \times 1$
conv3-1	conv3-2	$3 \times 3 \times 64 \times 64$	$64 \times 64 \times 64$	convt-F3	conv-R3	$3 \times 3 \times 64 \times 1$	$128 \times 128 \times 1$
conv3-2	conv3-3	$3 \times 3 \times 64 \times 64$	$64 \times 64 \times 64$	conv-I3 & conv-R3	HR-8×	element-wise sum	$128 \times 128 \times 1$
conv3-3	conv3-4	$3 \times 3 \times 64 \times 64$	$64 \times 64 \times 64$				
conv3-4	conv3-5	$3 \times 3 \times 64 \times 64$	$64 \times 64 \times 64$				
conv3-5	convt-F3	$4 \times 4 \times 64 \times 64$	$128 \times 128 \times 64$				

3. Additional Analysis

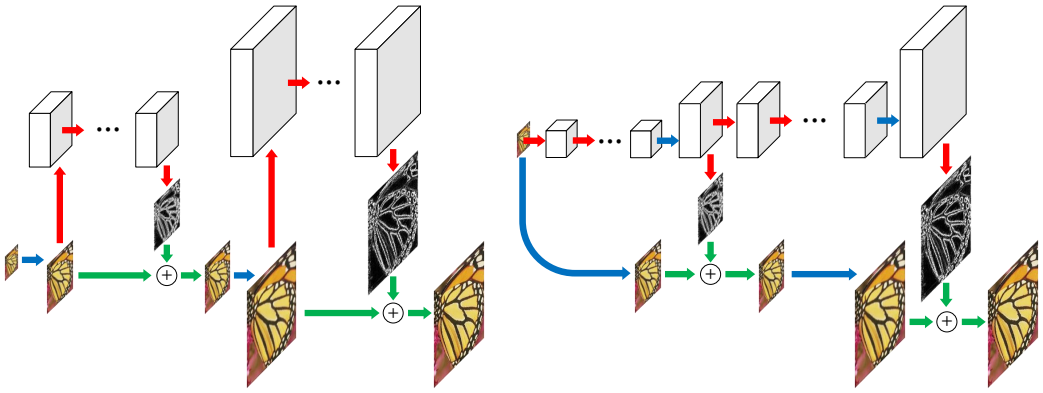
In this section, we first show quantitative and qualitative comparisons between the network architecture of LAPGAN [3] and the proposed LapSRN. We also extend the proposed method to incorporate the adversarial loss training. We then investigate the effect of using different training datasets for learning our model.

3.1. Comparison to the network architecture of LAPGAN

As described in the paper, the target applications of LAPGAN [3] and the proposed LapSRN are different. LAPGAN is a generative model that synthesizes diverse natural images from random noise and sample inputs, while our LapSRN is a super-resolution model that reconstructs accurate high-resolution images based on the input low-resolution images. Therefore, we focus on comparing the *network architectures* for image super-resolution here. In what follows, we denote "LAPGAN" as the generative network in [3].

Both LAPGAN and our LapSRN are inspired by the Laplacian pyramid framework. However, there are two main differences. First, LAPGAN upsamples input images *before* applying convolution at each level, while our LapSRN extracts features directly from the LR space and upscales images at the end of each level. Our network design effectively alleviates the computational cost and increases the size of receptive fields. Second, the sub-networks of LAPGAN are *independent*, while our LapSRN shares the feature representations from the lower scales to the finer scales. As a result, the residual images at a higher level are predicted by a deeper network. The feature sharing at lower levels increases the non-linearity and the network capacity to learn better mappings. We illustrate the architecture differences between LAPGAN and our LapSRN in Figure 1.

To investigate the effect of using different network architectures, we train LAPGAN and our LapSRN for 4× and 8× SR with the same training data and settings. We use 5 convolutional layers at each level and optimize both networks with the Charbonnier loss function. We note that in [3] the sub-networks are independently trained. To provide a fair comparison, we jointly train the entire network for LAPGAN and our LapSRN. Figure 2 shows the convergence curves (in terms of PSNR) for both networks on SET14. We provide quantitative comparisons of image quality and run time on SET5, SET14 and BSDS100 in Table 2. Under the same training setting, our method achieves more accurate reconstruction and faster speed than that of LAPGAN. Figure 3 shows visual comparisons on SET5 and SET14. Our LapSRN recovers accurate details for both 4× and 8× SR.

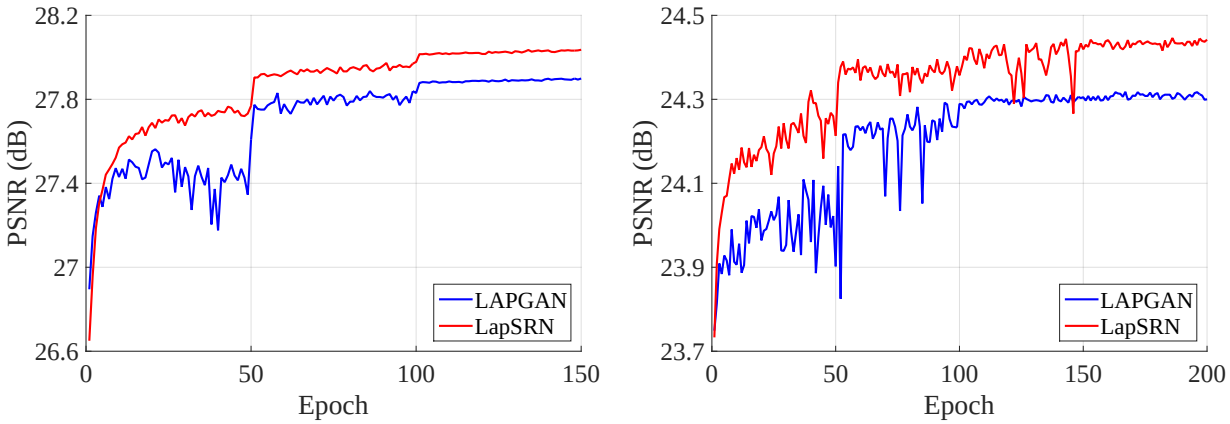


(a) Generative net of LAPGAN [3] (b) LapSRN (ours)

Figure 1: Network architectures of the generative net of LAPGAN [3] and the proposed LapSRN.

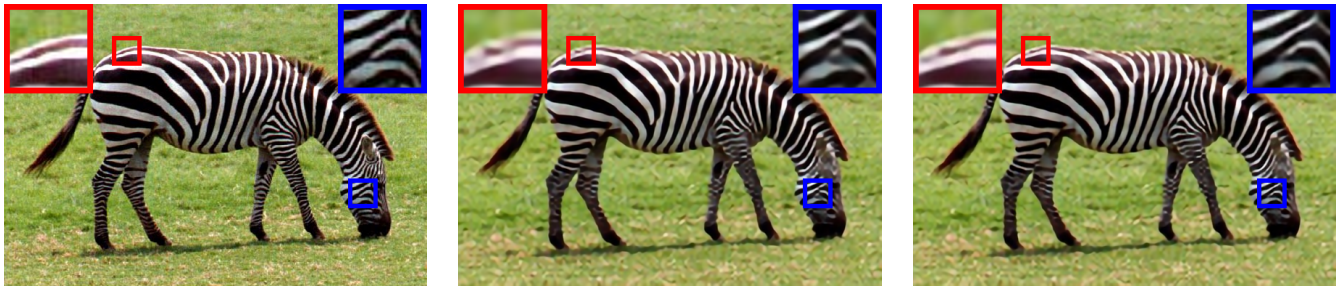
Table 2: Quantitative comparisons between the generative network of LAPGAN [3] and the proposed LapSRN. Our LapSRN achieves better quality and faster processing speed than LAPGAN.

Method	Scale	SET5			SET14			BSDS100		
		PSNR	SSIM	Seconds	PSNR	SSIM	Seconds	PSNR	SSIM	Seconds
LAPGAN	4	30.93	0.8744	0.0397	27.89	0.7633	0.0446	27.09	0.7195	0.0135
LapSRN	4	31.28	0.8800	0.0362	28.04	0.7675	0.0395	27.22	0.7238	0.0078
LAPGAN	8	25.85	0.7261	0.0478	24.30	0.6170	0.0518	24.46	0.5819	0.0110
LapSRN	8	26.09	0.7361	0.0376	24.42	0.6227	0.0427	24.53	0.5852	0.0107

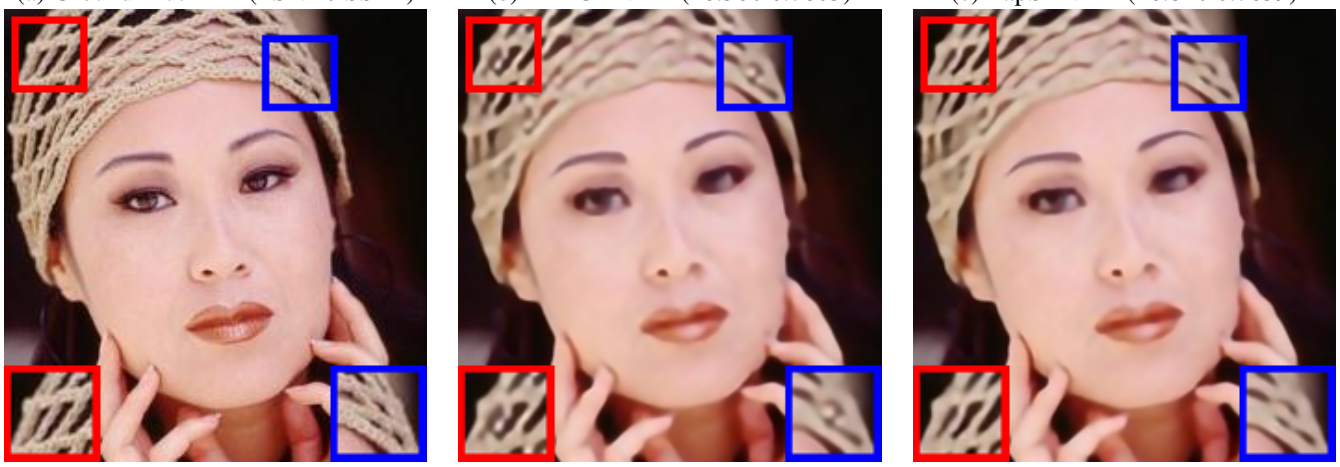


(a) 4× SR (b) 8× SR

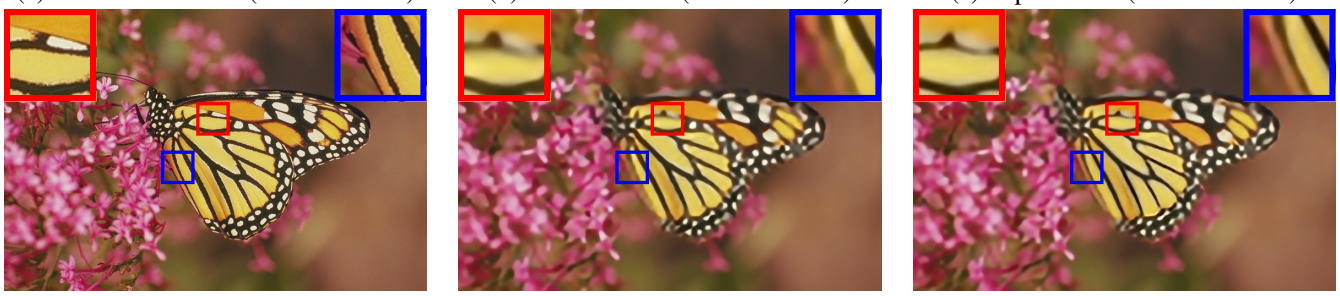
Figure 2: Convergence curves of LAPGAN and our LapSRN on SET14 for 4× and 8× SR, respectively.



(a) Ground Truth HR (PSNR / SSIM) (b) LAPGAN 4× (26.56 / 0.7605) (c) LapSRN 4× (26.84 / 0.7659)



(a) Ground Truth HR (PSNR / SSIM) (b) LAPGAN 4× (30.93 / 0.8744) (c) LapSRN 4× (31.28 / 0.8800)



(a) Ground Truth HR (PSNR / SSIM) (b) LAPGAN 8× (24.67 / 0.8142) (c) LapSRN 8× (24.81 / 0.8203)

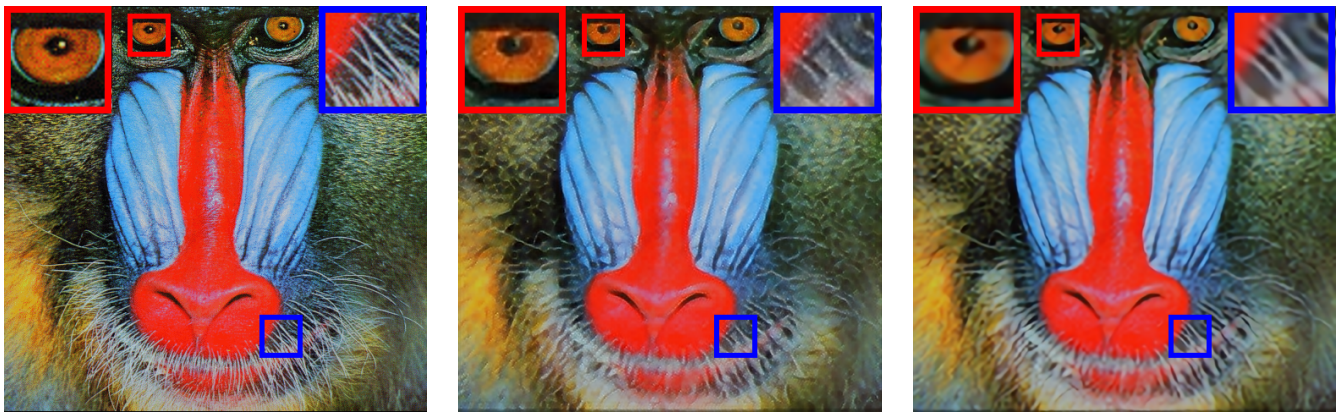
Figure 3: Visual comparison between LAPGAN and the proposed LapSRN on SET5 and SET14.

3.2. Adversarial training

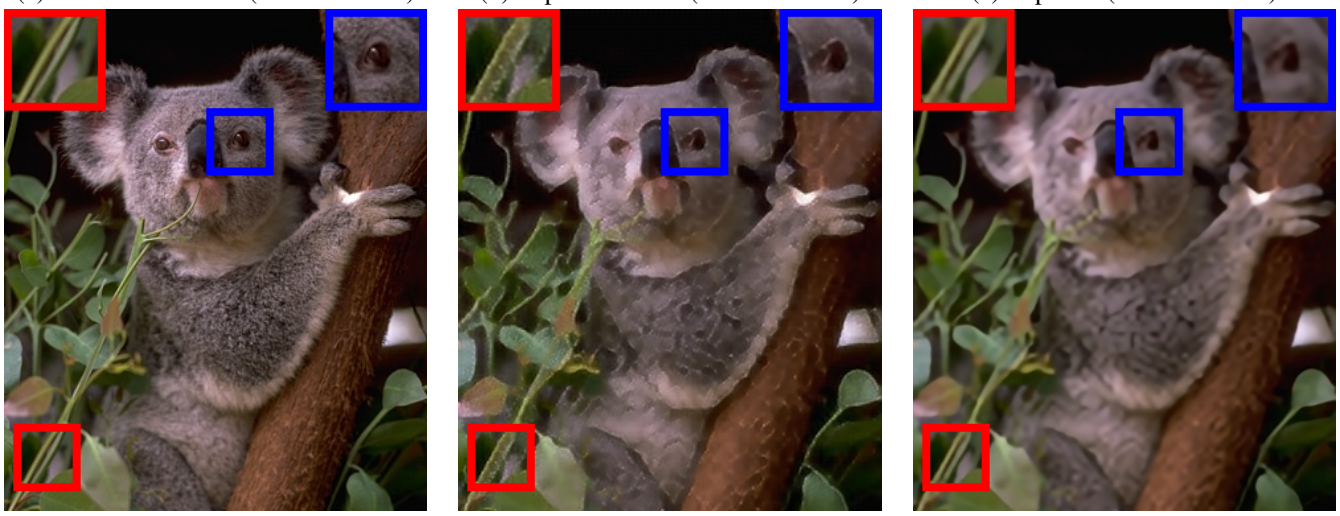
In [3], training the Generative Adversarial Network (GAN) consists of optimizing two networks: a generative network and a discriminative network. The generative network synthesizes realistic images from noise and sample inputs. The discriminative network learns to distinguish between the real images and the sampled images drawn from the generative network. The training is done by alternatively minimizing the cross-entropy loss functions. We refer readers to [3, 6] for more details. Similar training procedure has been applied to face images super-resolution [14], which demonstrates promising results on aligned face images for $8\times$ SR.

We demonstrate that the proposed LapSRN can be extended to incorporate the adversarial training. We treat our LapSRN as a generative network and design a discriminative network which takes an image with a spatial resolution 128×128 as input. The discriminative network consists of four convolutional layers with a stride of 2, two fully connected layers and one sigmoid layer to generate a scalar probability for distinguishing between real images and generated images from the generative network. We find that it is difficult to obtain accurate SR images by minimizing the cross-entropy loss functions only. As a result, we include the pixel-wise reconstruction loss function to enforce the similarity between the input LR images and the corresponding ground truth HR images. The overall loss function is a summation of the Charbonnier loss and the cross-entropy loss.

We show two visual results in Figure 4 for $4\times$ SR. The network with the adversarial training produces more realistic results on regions of irregular structures, e.g., hairs, grass, and feathers. However, the predicted results may not be faithful with respect to the ground truth high-resolution images.



(a) Ground Truth HR (PSNR / SSIM) (b) LapSRN + adv. (22.36 / 0.4754) (c) LapSRN (22.82 / 0.5147)



(a) Ground Truth HR (PSNR / SSIM) (b) LapSRN + adv. (28.87 / 0.7501) (c) LapSRN (30.41 / 0.8060)

Figure 4: Visual comparison between results trained with and without the adversarial training on $4\times$ SR.

3.3. Training datasets

Table 3 provides the implementation details of the state-of-the-art methods and the proposed LapSRN. We note that different approaches use different datasets for training. It is thus unclear which datasets are more suitable for training SR models. Does using more training data always improve the performance? In this section, we aim at validating the effect of three commonly used datasets in SR: T91 [13], BSDS200 [1] and General100 [5].

We train the proposed LapSRN (using depth $d = 5$, $4\times$ SR) with the following combinations of training datasets: (1) T91, (2) T91 + BSDS200, (3) T91 + General100 and (4) T91 + BSDS200 + General100. We plot the convergence curves on SET14 in Figure 5 and provide the quantitative results in Table 4. The quantitative results show that training with the 91 images from T91 already provides decent performance. Training with an additional set of images in BSDS200 does not improve the performance on most datasets except the BSDS100 testing set. Training with T91 and General100 leads to the best performance. We attribute the improvement to the fact that the T91 and General100 datasets contain images with sharp edges and less textured regions, which are suitable for learning SR models. We note that using more training data does not guarantee to obtain better performance since it may increase the difficulty for networks to fit the training data.

Table 3: Implementation details of the evaluated algorithms and the proposed approach.

Algorithm	Implementation	CPU / GPU	Training Dataset
A+ [11]	MATLAB	CPU	T91 [13]
SCN [12]	MATLAB	CPU	T91 [13]
DRCN [9]	MATLAB	GPU	T91 [13]
SRCNN [4]	MATLAB	GPU	ImageNet [2]
FSRCNN [5]	MATLAB	GPU	T91 [13] + General100 [4]
SelfExSR [7]	MATLAB	CPU	None
RFL [10]	MATLAB	CPU	T91 [13] + BSDS200 [1]
VDSR [8]	MATLAB	GPU	T91 [13] + BSDS200 [1]
DRCN [9]	MATLAB	GPU	T91 [13]
LapSRN (ours)	MATLAB	GPU	T91 [13] + BSDS200 [1]

Table 4: Quantitative comparisons between the networks trained on different datasets.

Training dataset	SET5		SET14		BSDS100		URBAN100		MANGA100	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
T91	31.29	0.880	28.04	0.767	27.18	0.722	24.95	0.746	28.59	0.880
T91 + BSDS200	31.28	0.880	28.04	0.768	**27.22**	**0.724**	25.01	0.747	28.64	0.882
T91 + General100	**31.32**	**0.880**	**28.10**	**0.768**	27.21	0.723	**25.04**	**0.748**	**28.79**	**0.883**
T91 + BSDS200 + General100	31.20	0.879	28.07	0.768	27.21	0.723	25.01	0.745	28.69	0.882

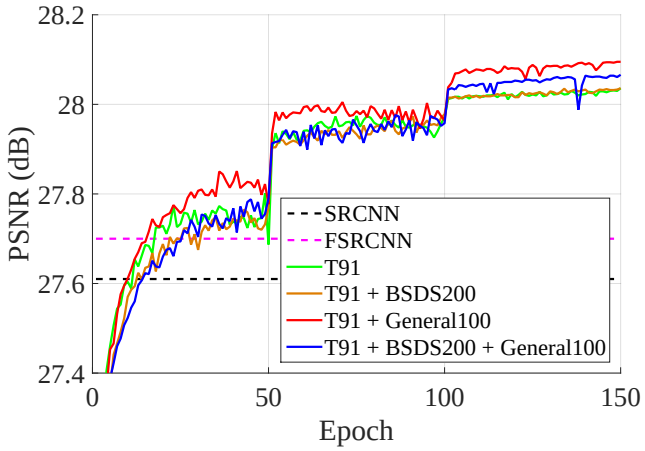


Figure 5: Convergence curves of networks trained on different datasets.

4. Runtime Evaluation

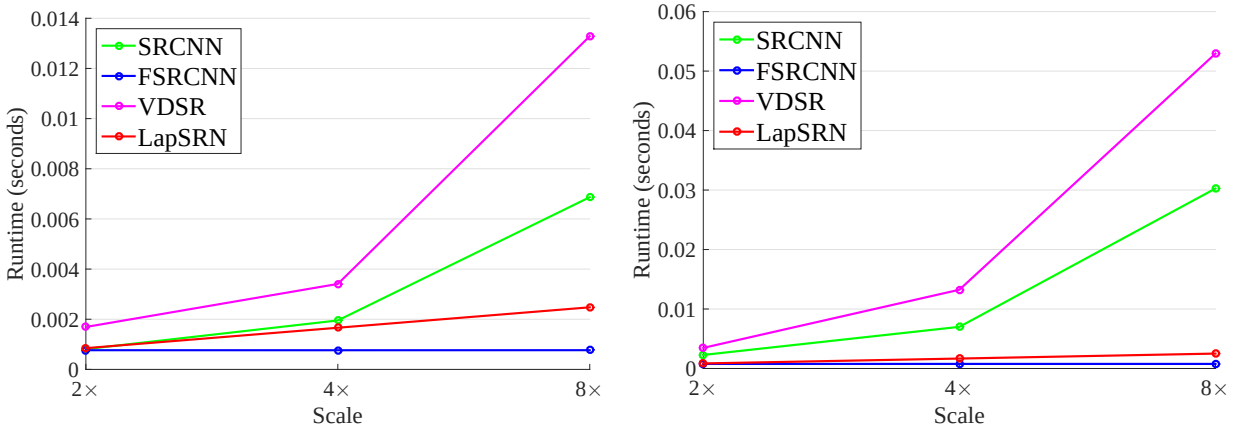
We provide quantitative comparisons regarding the execution time with 8 state-of-the-art SR algorithms: A+ [11], SRCNN [4], FSRCNN [5], SelfExSR [7], RFL [10], SCN [12], VDSR [8] and DRCN [9].

We evaluate the execution time of each algorithm on a machine with 3.4 GHz Intel i7 CPU (64G RAM) and Nvidia Titan X GPU (12G Memory). We take three LR images with spatial resolutions of 64×64, 128×128 and 256×256, and upscale by $2\times$, $4\times$ and $8\times$, respectively. We test each image for 10 times and report the averaged runtime in Table 5. In Figure 6, we focus on comparisons between CNN-based methods: SRCNN, FSRCNN, VDSR and our LapSRN. FSRCNN is the fastest algorithm since it applies convolution on LR images and has fewer network parameters. The runtime of SRCNN and VDSR depends on the size of *output* images, while the speed of FSRCNN is determined by the size of *input* images. The runtime of the proposed LapSRN mainly depends on the size of *input* images. Our LapSRN progressively upscales images and applies more convolutional layers on larger scaling factors. The time complexity increases slightly with respect to the target upsampling scales. However, the speed of LapSRN still performs favorably against SRCNN, VDSR and other existing algorithms.

Table 6 summarizes the average frame rate (FPS) on SET5, SET14, BSDS100, URBAN100 and MANGA109 with the scale factors $2\times$, $4\times$ and $8\times$, respectively. Both FSRCNN [5], and our LapSRN achieve real-time speed (i.e., 24 frames per second) on most datasets. While our network contains more convolutional layers than that in SRCNN, SCN and VDSR, we super-resolve images efficiently due to the design of the pyramidal structure.

Table 5: Runtime evaluation with different sizes of *input* images. We note that SRCNN, VDSR, DRCN and our LapSRN run out of the GPU memory when upsampling the image with the spatial resolution 256×256 for $8\times$ (the size of output images is 2048×2048).

Input size		64			128			256	
Output size	128	256	512	256	512	1024	512	1024	2048
Scale factor	$2\times$	$4\times$	$8\times$	$2\times$	$4\times$	$8\times$	$2\times$	$4\times$	$8\times$
A+ [11]	0.01139	0.01656	0.03474	0.04646	0.06581	0.18079	0.19340	0.32775	1.67798
SRCNN [4]	0.00081	0.00195	0.00687	0.00228	0.00700	0.03026	0.00675	0.02988	N.A.
FSRCNN [5]	**0.00076**	**0.00076**	**0.00077**	**0.00077**	**0.00077**	**0.00077**	**0.00077**	**0.00077**	0.00078
RFL [10]	0.02546	0.03164	0.07021	0.05256	0.10095	0.49087	0.16088	0.59435	2.61875
SCN [12]	0.01526	0.09404	0.34208	0.05806	0.35590	1.27679	0.27957	1.25322	6.08012
VDSR [8]	0.00169	0.00341	0.01329	0.00347	0.01324	0.05294	0.01326	0.05278	N.A.
DRCN [9]	0.01704	0.06620	0.10131	0.06520	0.31282	0.30750	0.30909	1.15490	N.A.
LapSRN (ours)	0.00085	0.00166	0.00248	0.00085	0.00167	0.00251	0.00085	0.00168	N.A.



(a) Fix size of input images to 64×64

(b) Fix size of input images to 128×128

Figure 6: Trade-off between runtime and size of *input* images. We fix the size of input images and perform $2\times$, $4\times$ and $8\times$ SR with SRCNN [4], FSRCNN [5], VDSR [8] and the proposed LapSRN, respectively.

Table 6: Comparison of the FPS (frames per second) on the 5 benchmark datasets with scale factors 2×, 4× and 8×. **Red color** indicates the fastest algorithm and blue color indicates the second fastest algorithm. Both FSRCNN and our method achieve real-time performance (i.e., 24 FPS) on most datasets.

Algorithm	Scale	SET5	SET14	BSDS100	URBAN100	MANGA109
A+ [11]	2	1.12	0.52	0.74	0.15	0.10
SRCNN [4]	2	24.70	22.92	39.50	9.03	6.53
FSRCNN [5]	2	**31.04**	**53.86**	**98.20**	**47.23**	34.48
SelfExSR [7]	2	0.02	0.01	0.01	0.00	0.00
RFL [10]	2	0.65	0.45	0.52	0.13	0.15
SCN [12]	2	1.19	0.85	1.19	0.24	0.17
VDSR [8]	2	11.01	6.46	10.00	2.12	1.71
DRCN [9]	2	0.70	0.37	0.59	0.10	0.08
LapSRN (ours)	2	30.20	40.00	97.36	16.81	**85.32**
A+ [11]	4	2.86	1.62	2.43	0.49	0.41
SRCNN [4]	4	21.74	22.27	40.13	9.95	7.13
FSRCNN [5]	4	**31.61**	**56.58**	**101.54**	**53.95**	**55.23**
SelfExSR [7]	4	0.04	0.02	0.03	0.00	0.00
RFL [10]	4	1.97	1.21	1.64	0.42	0.34
SCN [12]	4	1.38	0.87	1.19	0.31	0.25
VDSR [8]	4	10.71	6.59	9.91	2.15	1.76
DRCN [9]	4	0.80	0.37	0.59	0.10	0.08
LapSRN (ours)	4	25.49	25.46	54.35	12.40	47.63
A+ [11]	8	5.79	2.84	4.31	0.80	0.64
SRCNN [4]	8	20.92	17.69	40.13	9.81	7.17
FSRCNN [5]	8	**34.10**	**63.28**	**104.46**	**71.67**	**71.95**
SelfExSR [7]	8	0.03	0.01	0.02	0.00	0.00
RFL [10]	8	2.54	1.61	2.25	0.47	0.33
SCN [12]	8	0.79	0.53	0.68	0.21	0.19
VDSR [8]	8	10.58	6.50	10.13	2.15	1.77
LapSRN (ours)	8	24.02	23.40	50.44	10.54	33.09

5. More Quantitative Comparisons

Due to the network design of the proposed LapSRN, the scale factors for *training* are limited to the power of 2, e.g., 2×, 4× or 8×. However, our LapSRN can perform SR to arbitrary scales by first upsampling input images to a larger scale and then downsampling the output images to the desired resolution. The quantitative results for 2×, 4× and 8× SR are presented in the main paper. In Table 7, we show the numerical results for 3× SR by using our 4× model. We note that we do not use any 3× SR images to optimize our network. However, our 4× model provides comparable performance with state-of-the-art methods thanks to the deeply supervised training.

Table 7: Quantitative evaluation of state-of-the-art SR algorithms: average PSNR/SSIM/IFC for scale factors 3×. **Red** text indicates the best and blue text indicates the second best performance.

Algorithm	Scale	SET5 PSNR / SSIM / IFC	SET14 PSNR / SSIM / IFC	BSDS100 PSNR / SSIM / IFC	URBAN100 PSNR / SSIM / IFC	MANGA109 PSNR / SSIM / IFC
Bicubic	3	30.39 / 0.868 / 3.596	27.64 / 0.776 / 3.491	27.21 / 0.740 / 3.168	24.46 / 0.736 / 3.661	26.98 / 0.858 / 3.521
A+ [11]	3	32.60 / 0.908 / 4.979	29.24 / 0.821 / 4.545	28.30 / 0.784 / 4.028	26.05 / 0.798 / 4.883	29.91 / 0.911 / 4.880
SRCNN [4]	3	32.76 / 0.908 / 4.682	29.41 / 0.823 / 4.373	28.41 / 0.787 / 3.879	26.24 / 0.800 / 4.630	30.58 / 0.913 / 4.698
FSRCNN [5]	3	33.15 / 0.913 / 4.971	29.53 / 0.826 / 4.569	28.52 / 0.790 / 4.061	26.42 / 0.807 / 4.878	31.09 / 0.920 / 4.912
SelfExSR [7]	3	32.63 / 0.908 / 4.911	29.33 / 0.823 / 4.505	28.29 / 0.785 / 3.922	26.45 / 0.809 / 4.988	27.57 / 0.820 / 2.193
RFL [10]	3	32.45 / 0.905 / 4.956	29.15 / 0.819 / 4.532	28.22 / 0.782 / 4.023	25.87 / 0.791 / 4.781	29.60 / 0.904 / 4.758
SCN [12]	3	32.60 / 0.907 / 4.321	29.24 / 0.819 / 4.006	28.32 / 0.782 / 3.553	26.21 / 0.801 / 4.253	30.21 / 0.912 / 4.302
VDSR [8]	3	33.66 / 0.921 / 5.088	29.77 / **0.834** / 4.606	**28.83** / **0.798** / 4.043	27.14 / **0.829** / 5.045	31.99 / 0.933 / 5.389
DRCN [9]	3	**33.82** / **0.922** / **5.202**	29.76 / 0.833 / **4.686**	28.80 / 0.797 / **4.070**	**27.15** / 0.828 / **5.187**	**32.29** / **0.935** / **5.564**
LapSRN (ours)	3	33.78 / 0.921 / 5.194	**29.87** / 0.833 / 4.665	28.81 / 0.797 / 4.057	27.06 / 0.827 / 5.168	32.19 / 0.934 / 5.406

6. Qualitative Comparisons

In this section, we provide more visual comparisons with state-of-the-art methods on BSDS100, URBAN100, MANGA109 and the real-world historical photos. The complete results on all benchmark datasets for 2×, 4× and 8× SR are provided on our project website http://vllab1.ucmerced.edu/~wlai24/LapSRN.

6.1. Visual comparisons on BSDS100 with 4× SR

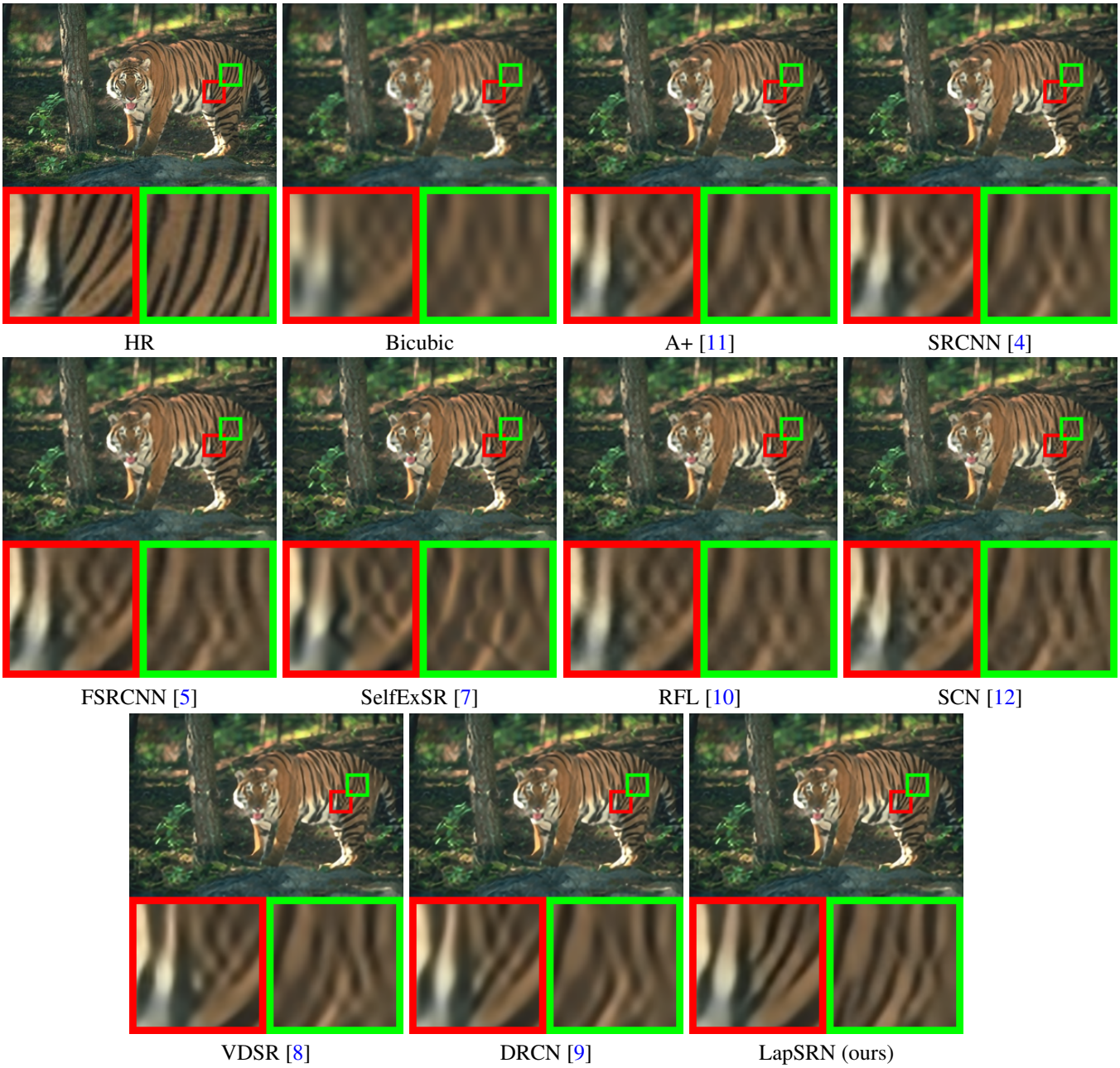


Figure 7: Visual comparison for 4× SR on BSDS100.

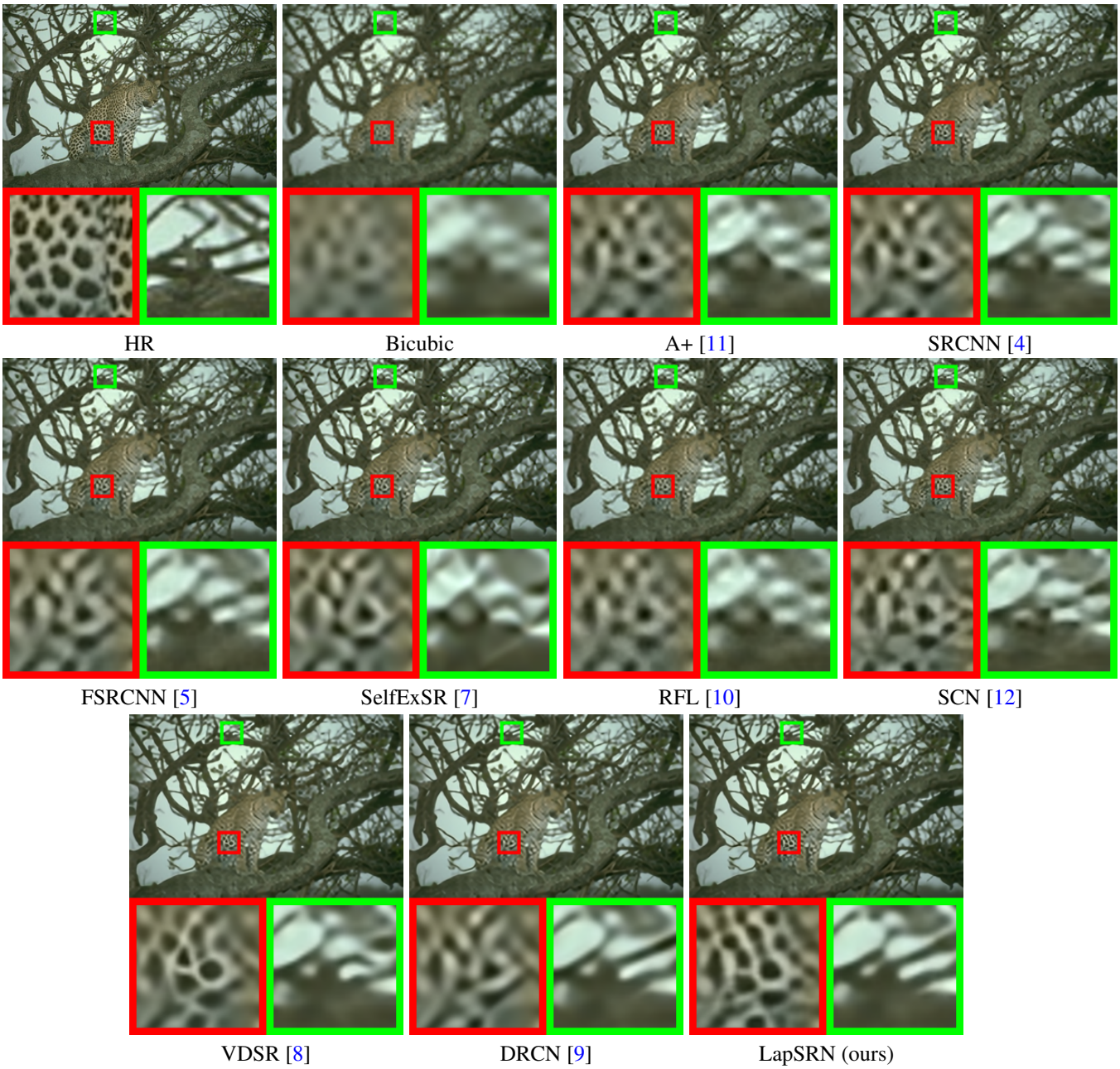
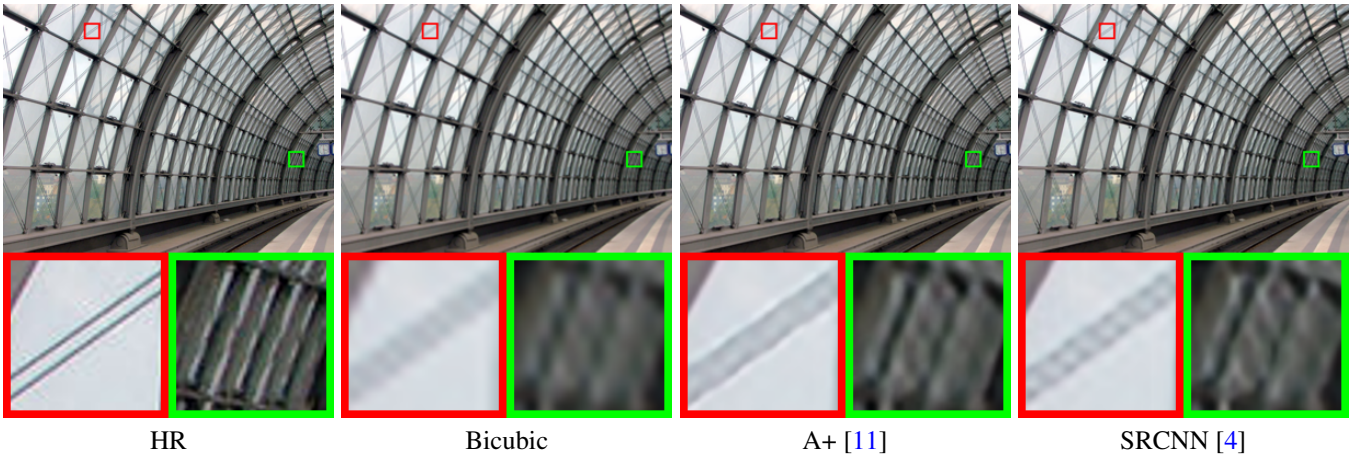


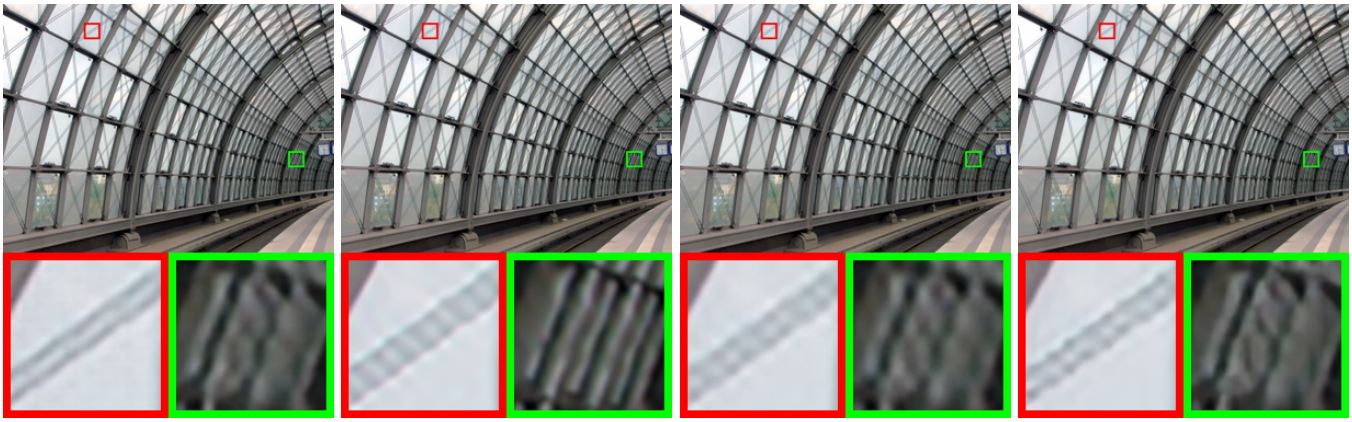
Figure 8: Visual comparison for 4× SR on BSDS100.

Figure 9: Visual comparison for 4× SR on BSDS100.

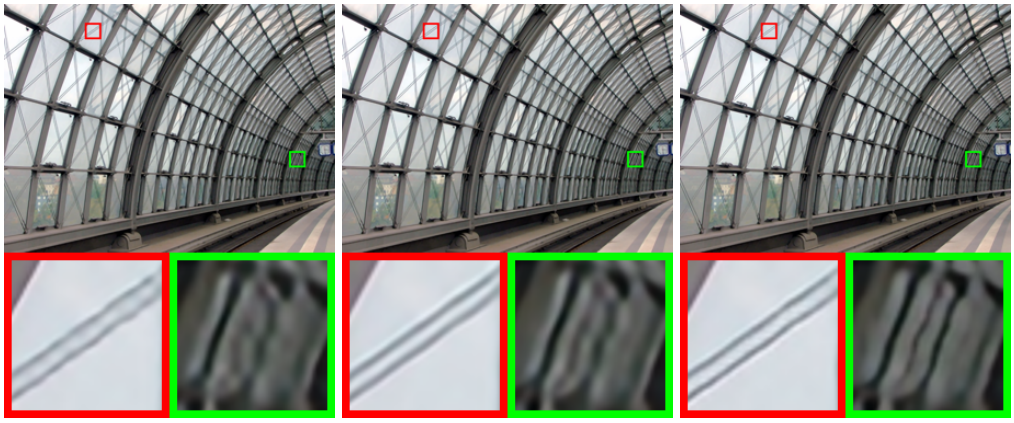
6.2. Visual comparisons on Urban100 with 4× SR



HR | Bicubic | A+ [11] | SRCNN [4]



FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]



VDSR [8] | DRCN [9] | LapSRN (ours)

Figure 10: Visual comparison for 4× SR on URBAN100.

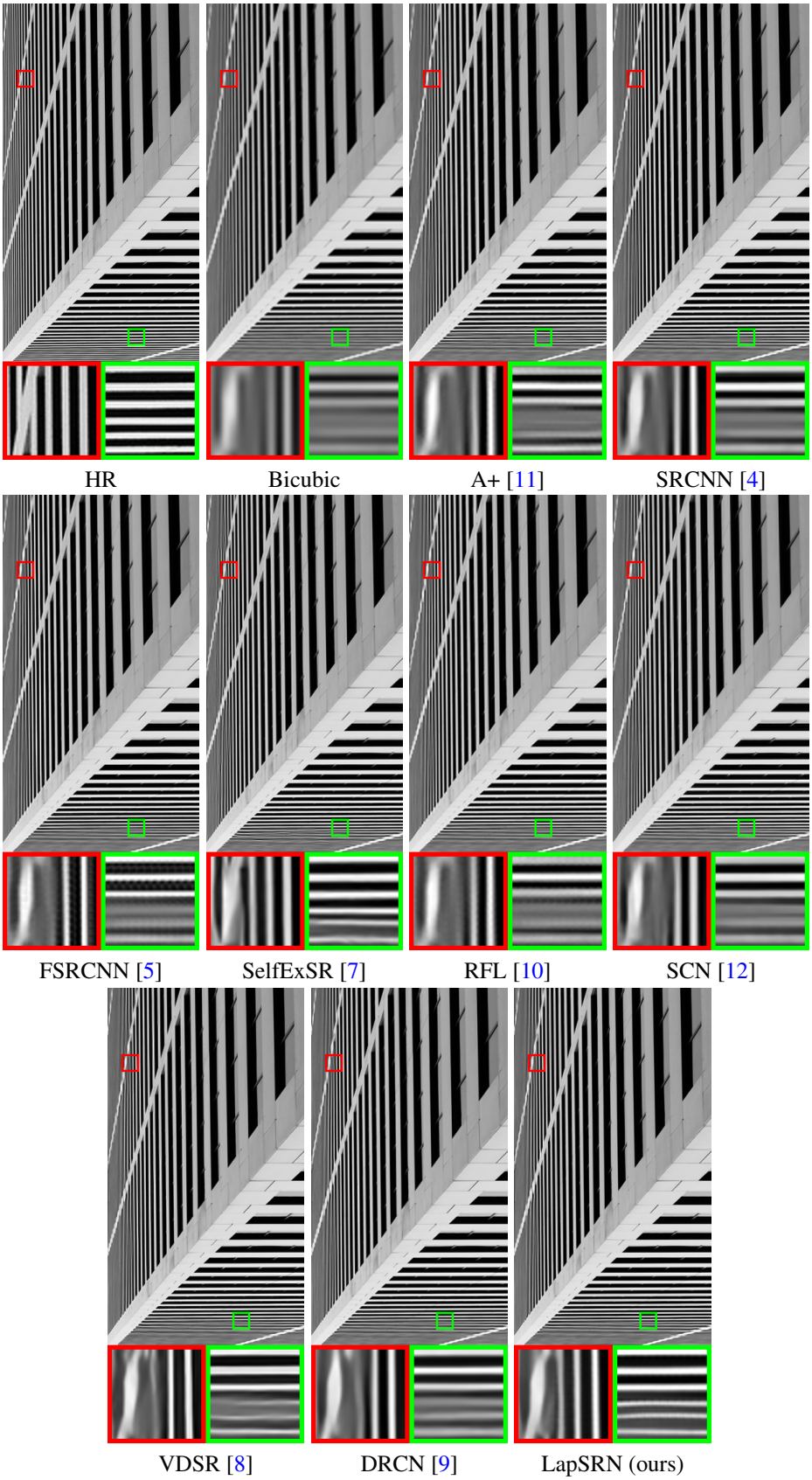


Figure 11: Visual comparison for 4× SR on URBAN100.

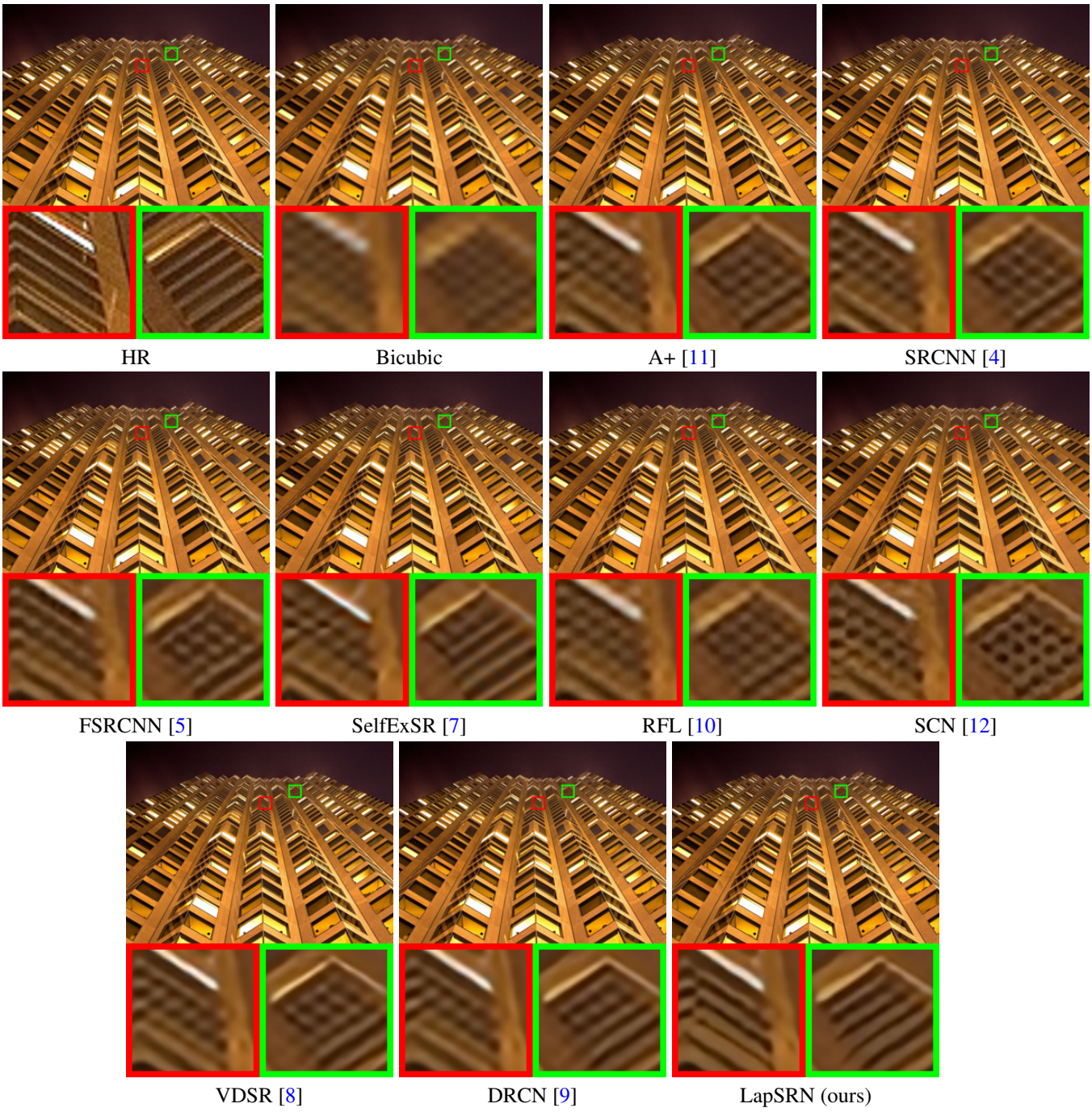


Figure 12: Visual comparison for 4× SR on URBAN100.

Figure 13: Visual comparison for 4× SR on URBAN100.

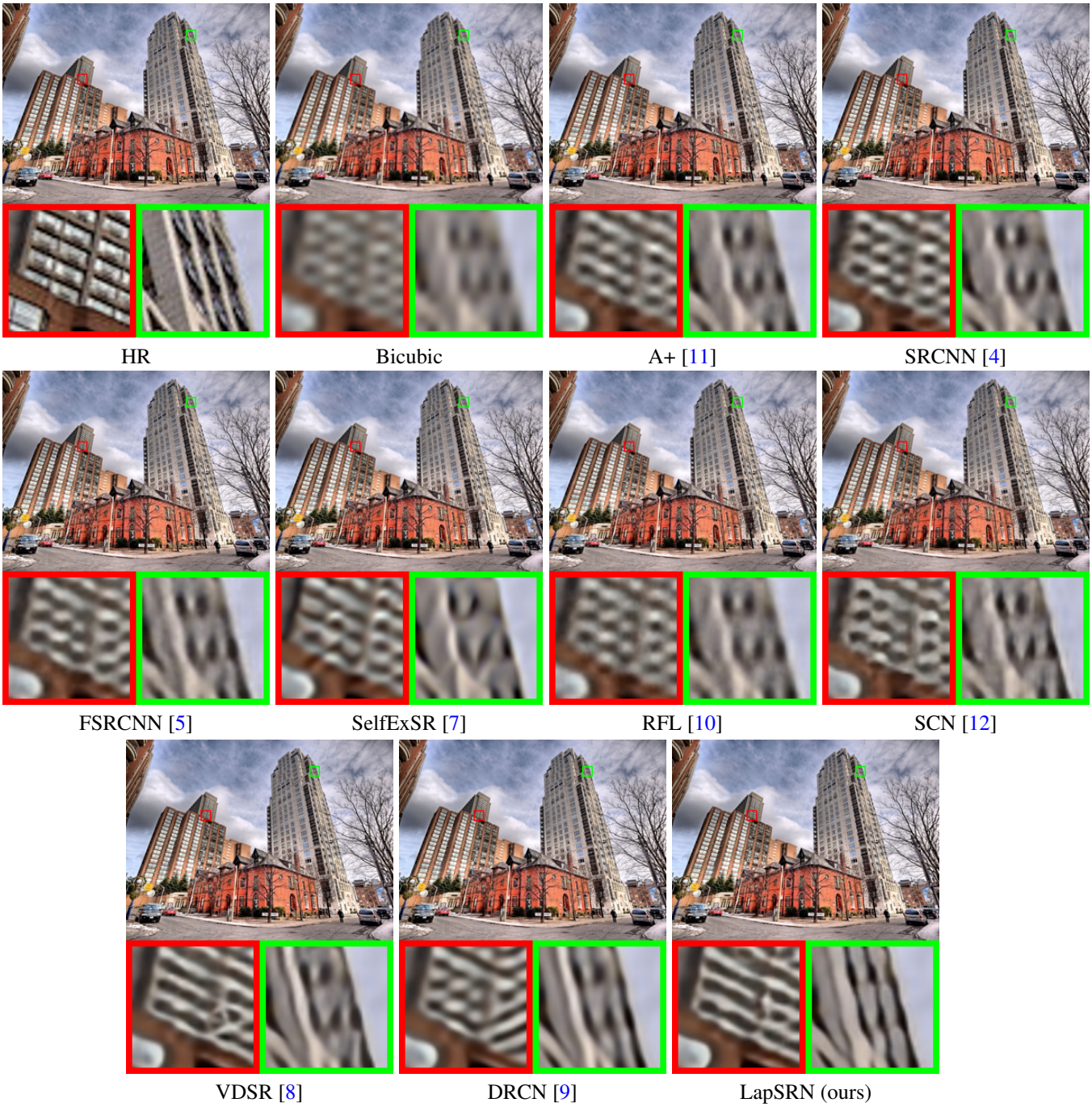


Figure 14: Visual comparison for 4× SR on URBAN100.

Figure 15: Visual comparison for 4× SR on URBAN100.

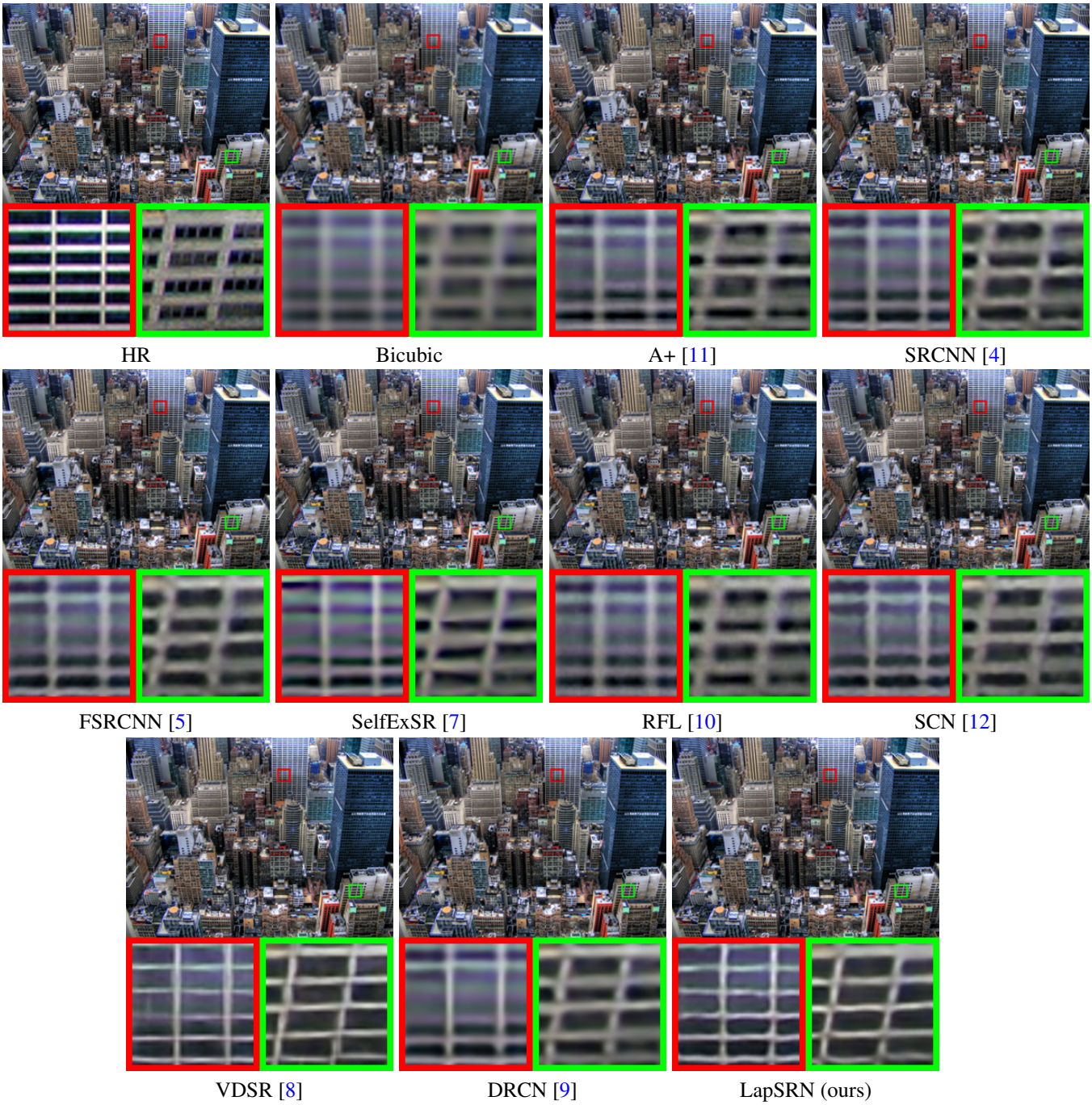


Figure 16: Visual comparison for 4× SR on URBAN100.

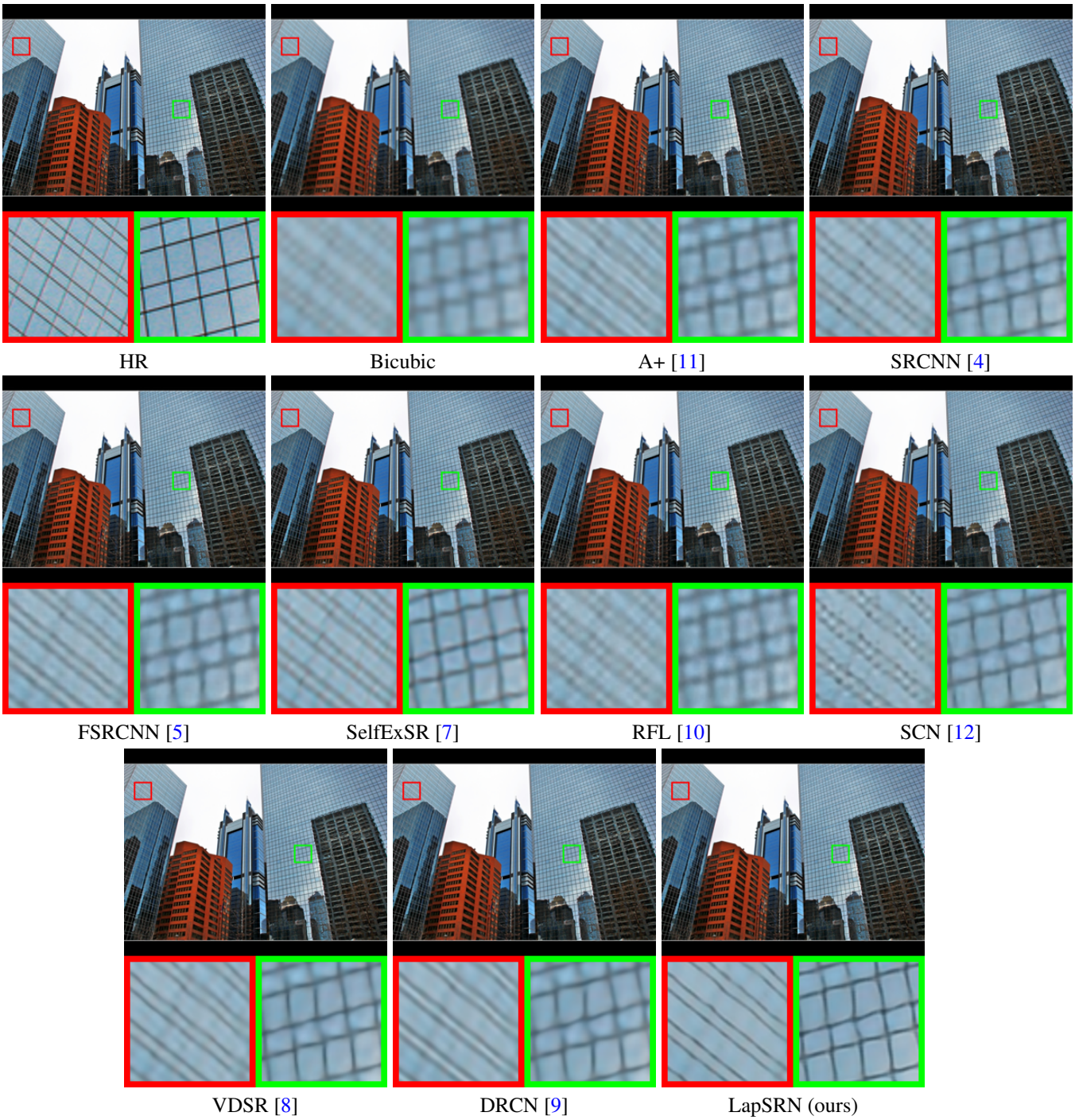


Figure 17: Visual comparison for 4× SR on URBAN100.

6.3. Visual comparisons on Manga109 with 4× SR

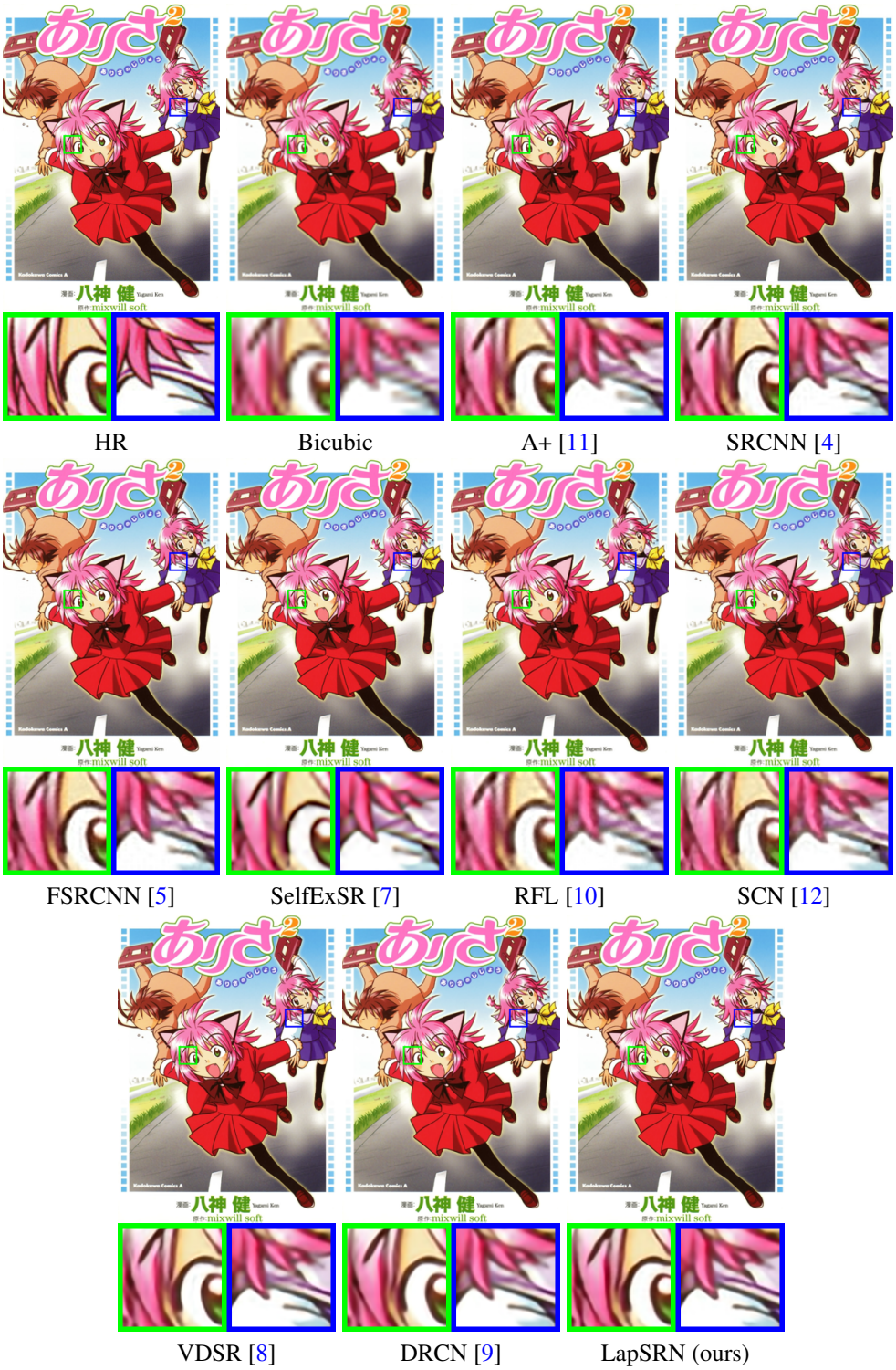


Figure 18: Visual comparison for 4× SR on MANGA109.

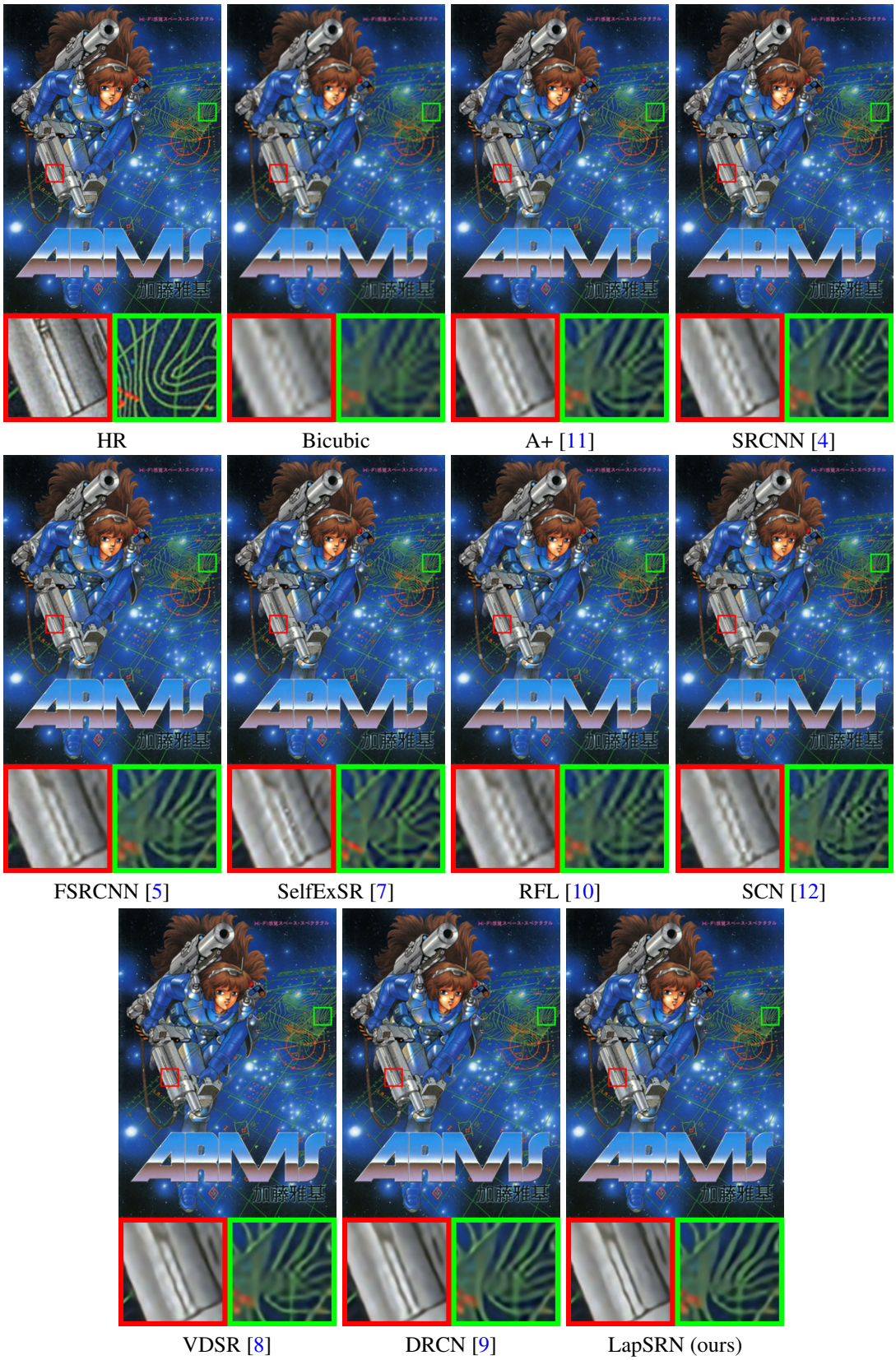


Figure 19: Visual comparison for 4× SR on MANGA109.

HR | Bicubic | A+ [11] | SRCNN [4]

FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

VDSR [8] | DRCN [9] | LapSRN (ours)

Figure 20: Visual comparison for 4× SR on MANGA109.

Figure 21: Visual comparison for 4× SR on MANGA109.

Figure 22: Visual comparison for 4× SR on MANGA109.

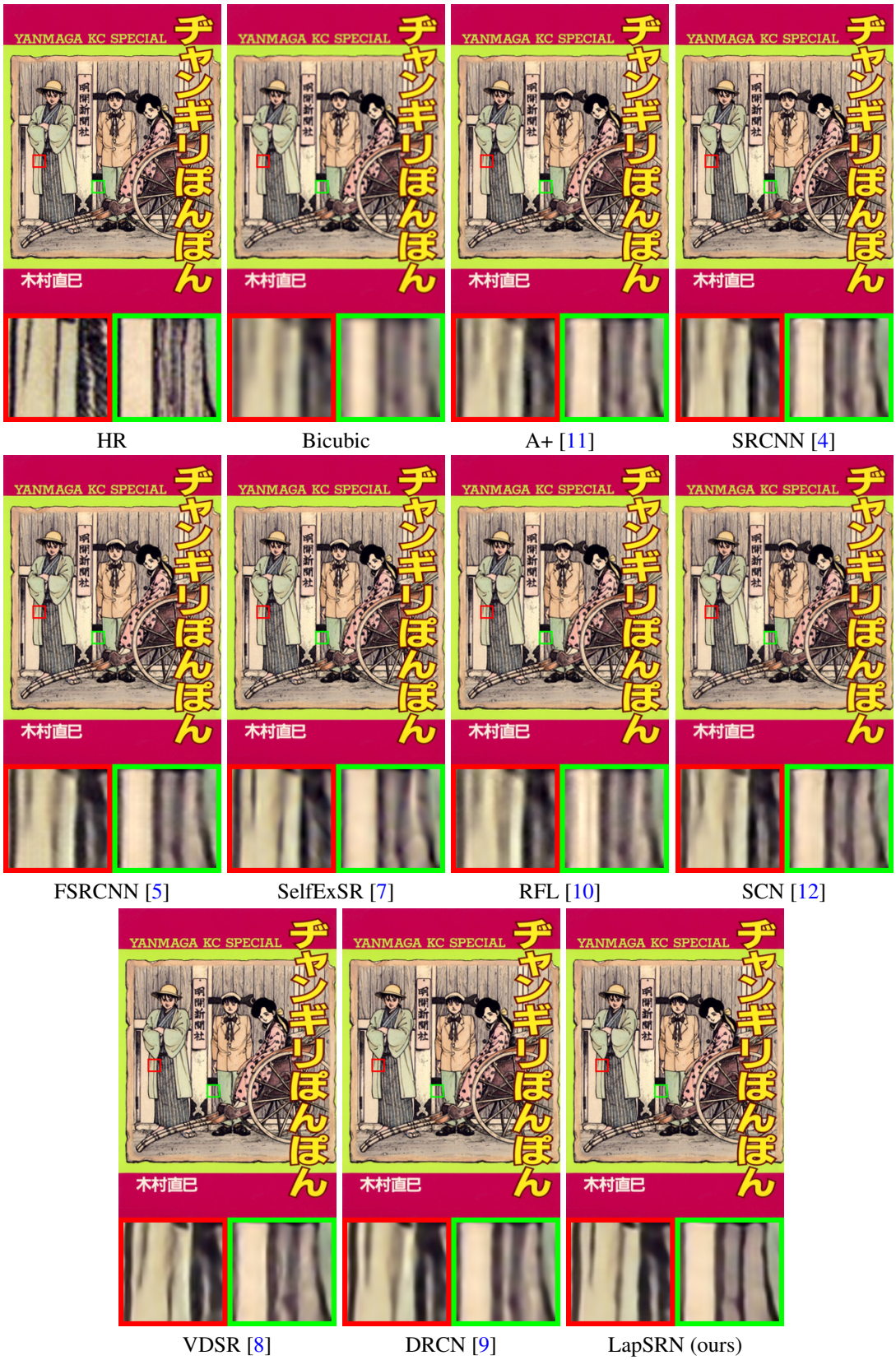


Figure 23: Visual comparison for 4× SR on MANGA109.

Figure 24: Visual comparison for 4× SR on MANGA109.

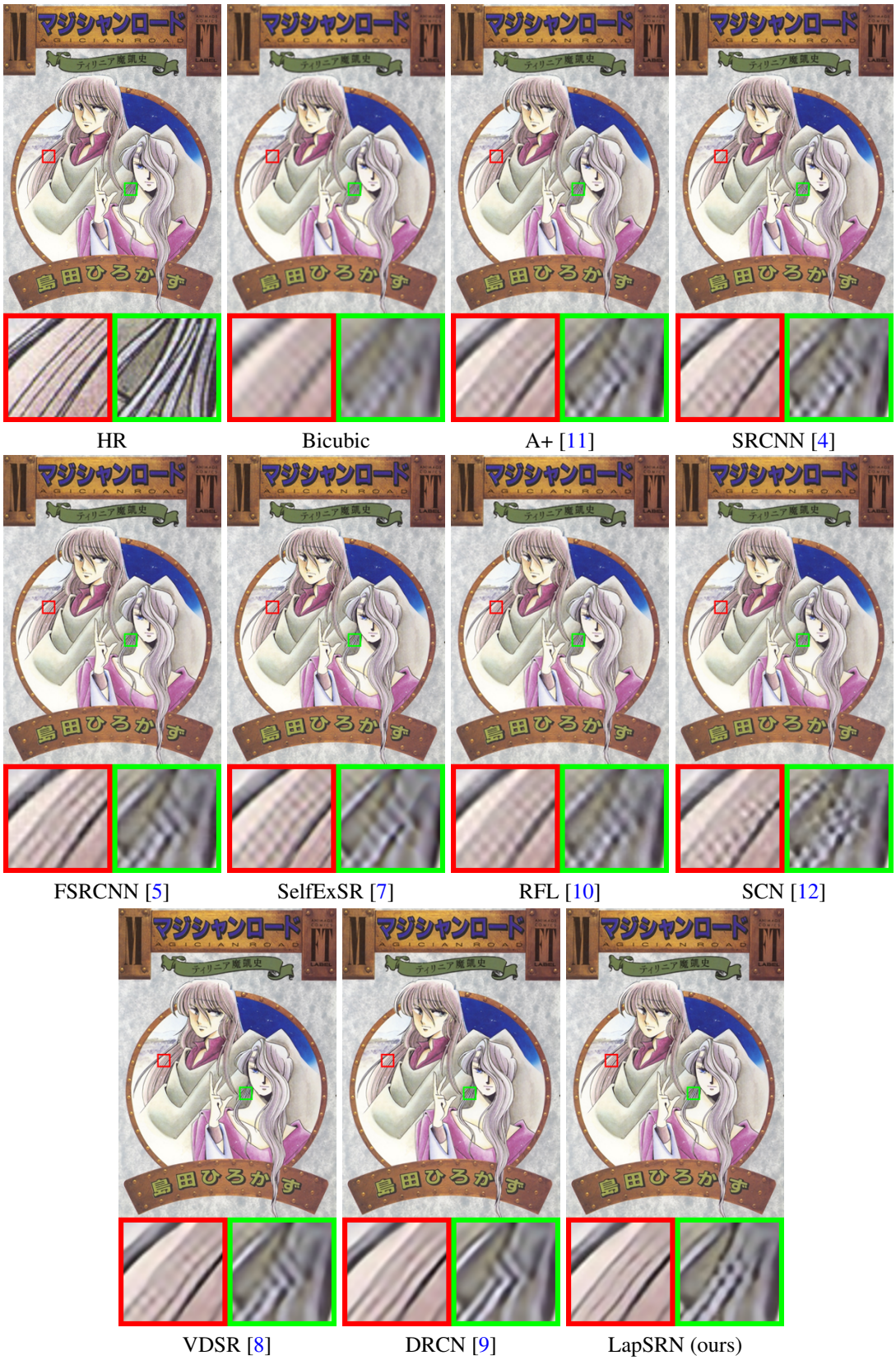


Figure 25: Visual comparison for 4× SR on MANGA109.

Figure 26: Visual comparison for 4× SR on MANGA109.

6.4. Visual comparisons on BSDS100 with 8× SR

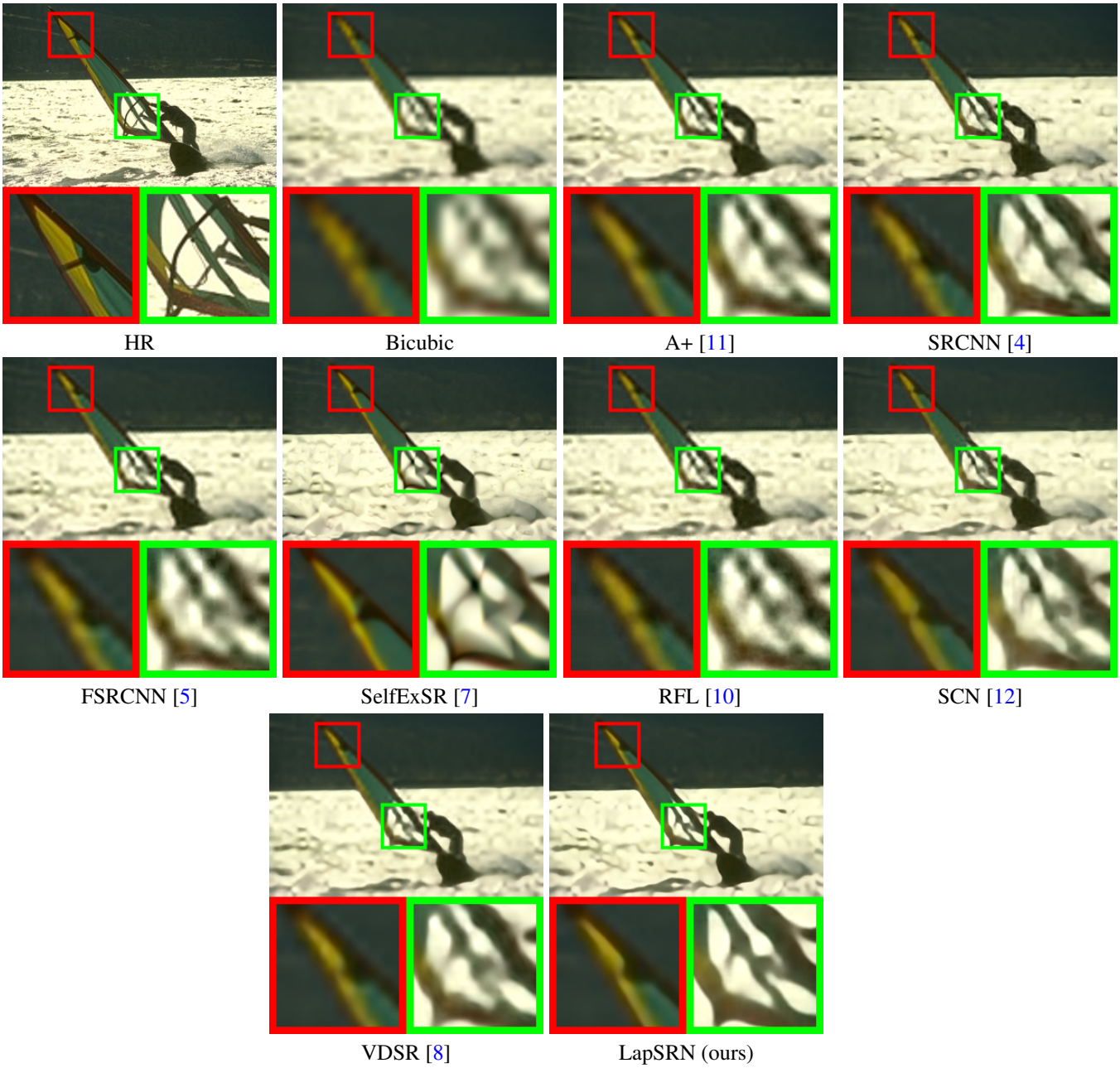


Figure 27: Visual comparison for 8× SR on BSDS100.

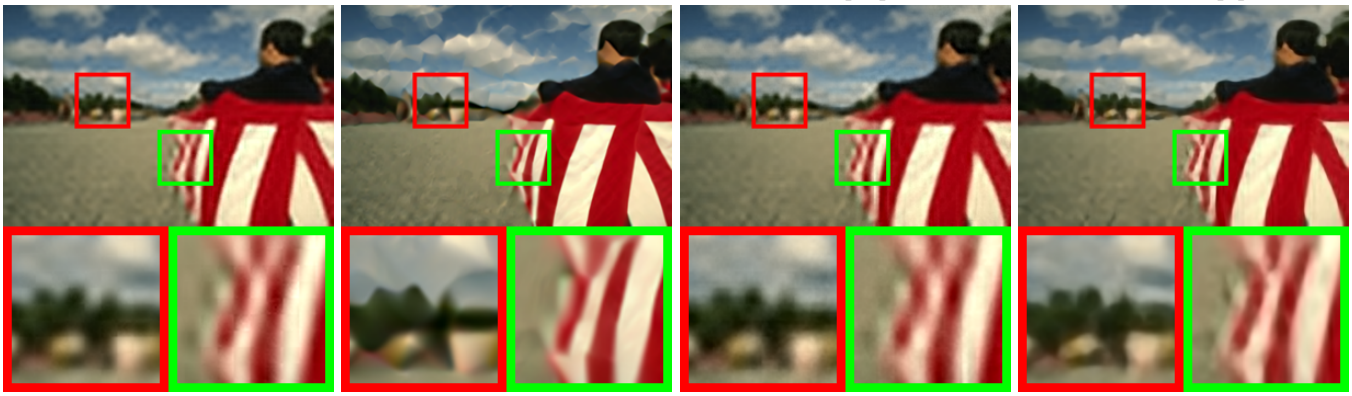
HR | Bicubic | A+ [11] | SRCNN [4]

FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

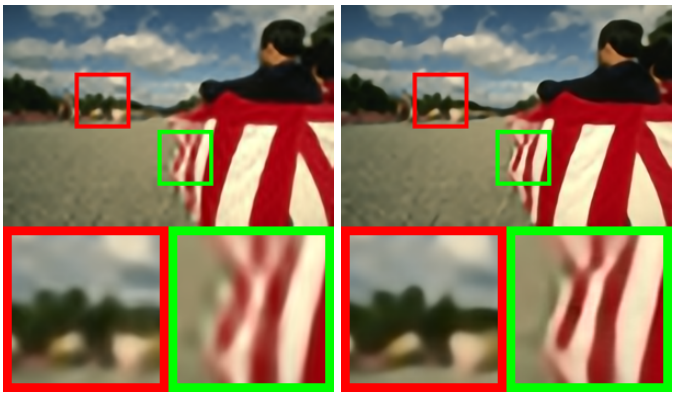
VDSR [8] | LapSRN (ours)

Figure 28: Visual comparison for 8× SR on BSDS100.

HR | Bicubic | A+ [11] | SRCNN [4]



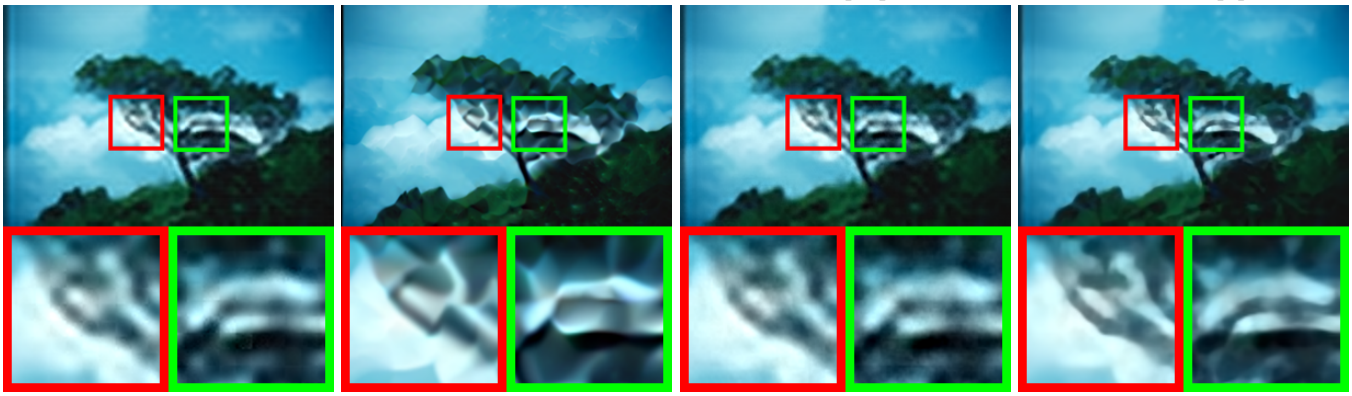
FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]



VDSR [8] | LapSRN (ours)

Figure 29: Visual comparison for 8× SR on BSDS100.

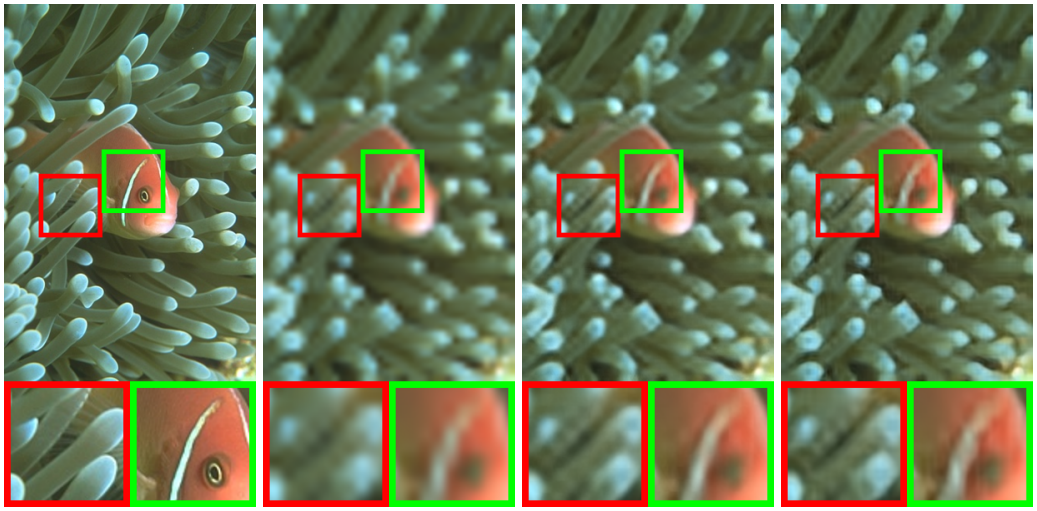
HR | Bicubic | A+ [11] | SRCNN [4]



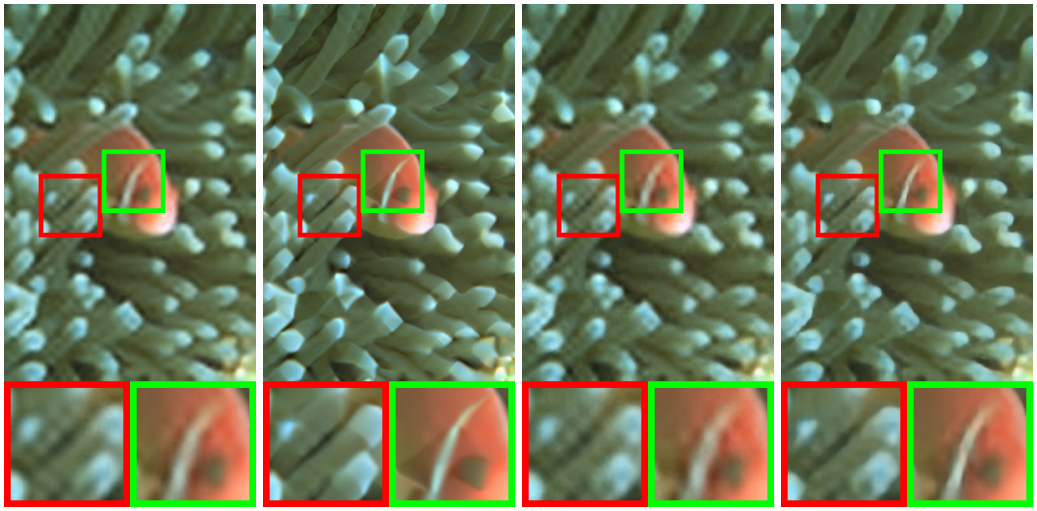
FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

VDSR [8] | LapSRN (ours)

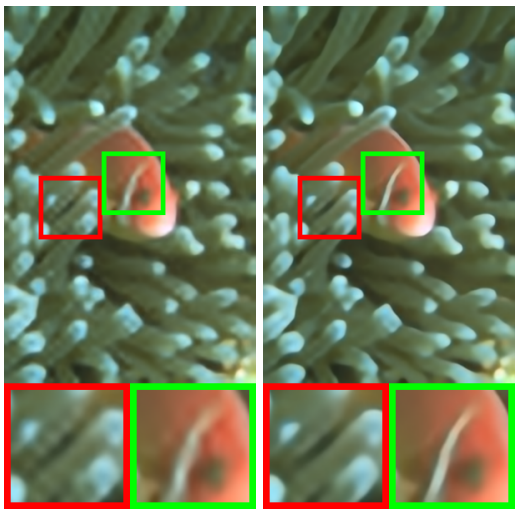
Figure 30: Visual comparison for 8× SR on BSDS100.



HR | Bicubic | A+ [11] | SRCNN [4]



FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]



VDSR [8] | LapSRN (ours)

Figure 31: Visual comparison for 8× SR on BSDS100.

HR | Bicubic | A+ [11] | SRCNN [4]

FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

VDSR [8] | LapSRN (ours)

Figure 32: Visual comparison for 8× SR on BSDS100.

6.5. Visual comparisons on Urban100 with 8× SR

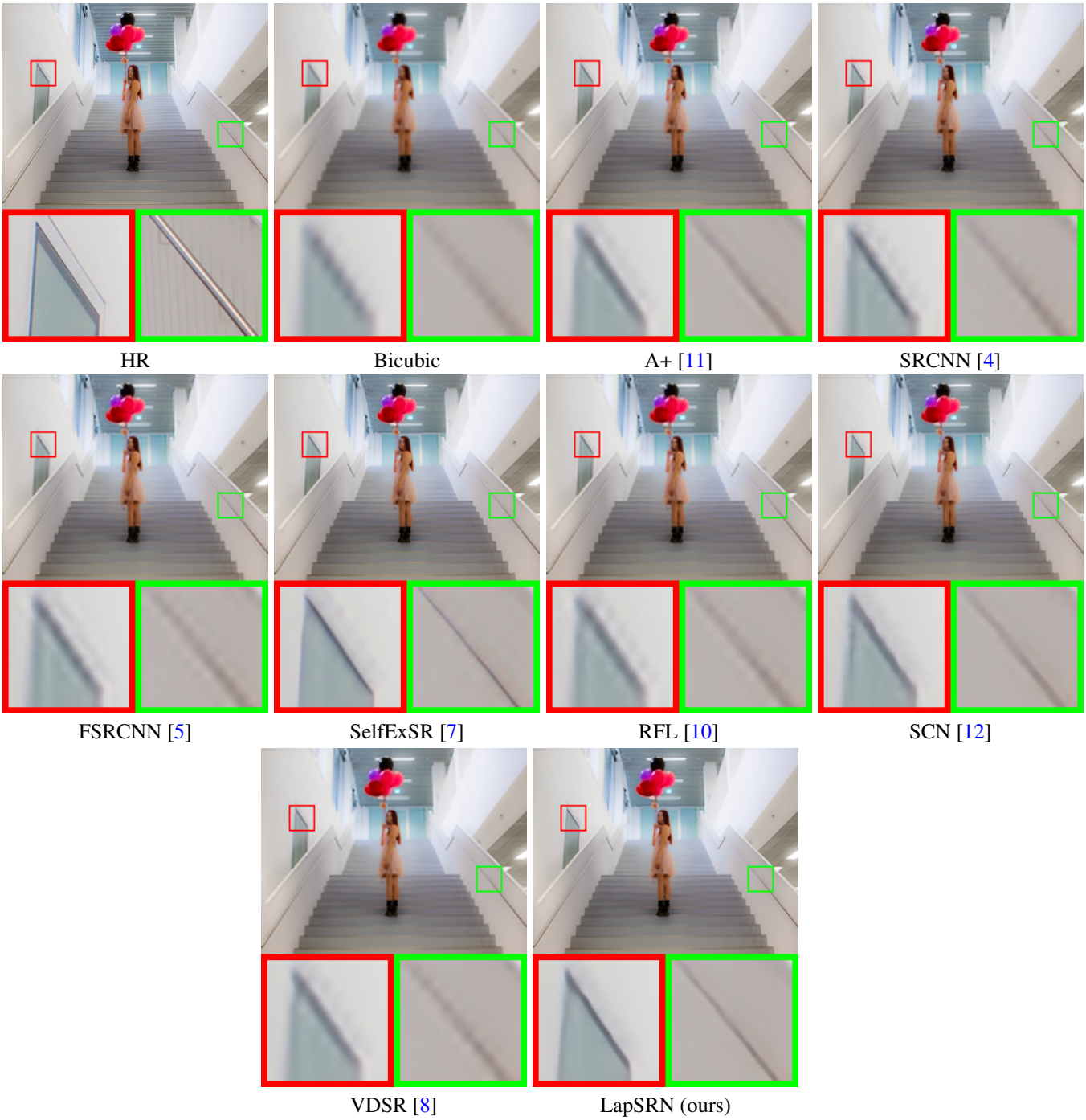
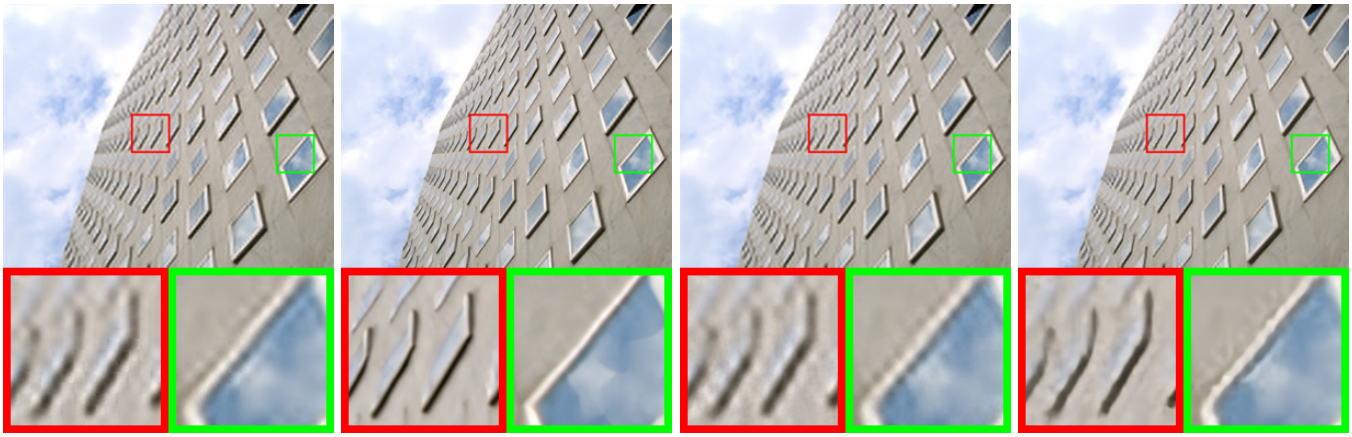


Figure 33: Visual comparison for 8× SR on URBAN100.

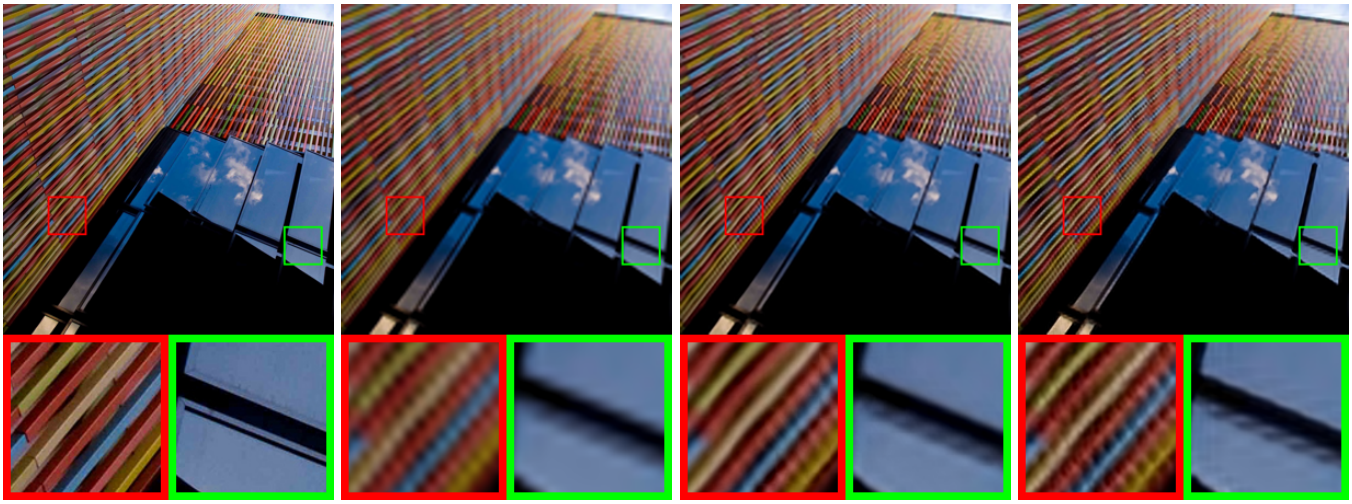
HR | Bicubic | A+ [11] | SRCNN [4]



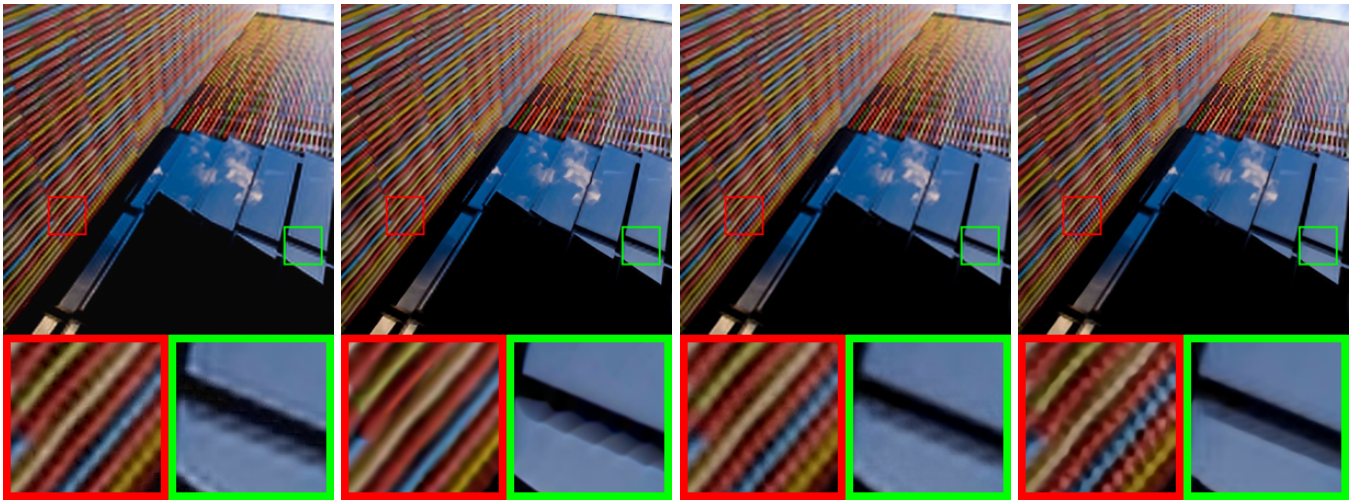
FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

VDSR [8] | LapSRN (ours)

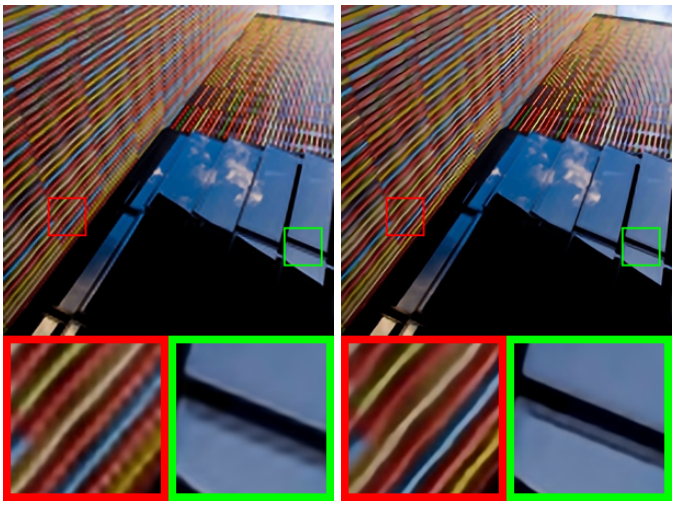
Figure 34: Visual comparison for 8× SR on URBAN100.



HR | Bicubic | A+ [11] | SRCNN [4]

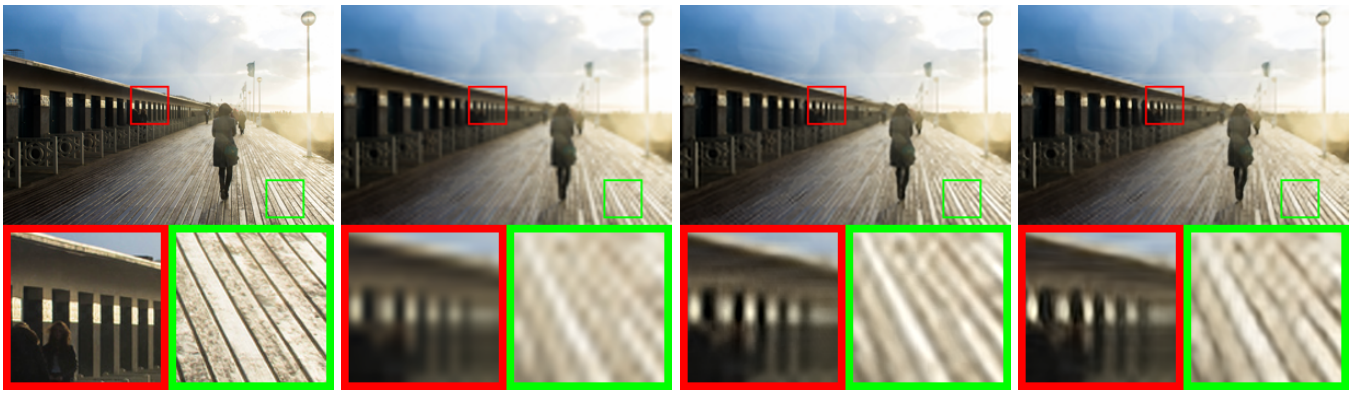


FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

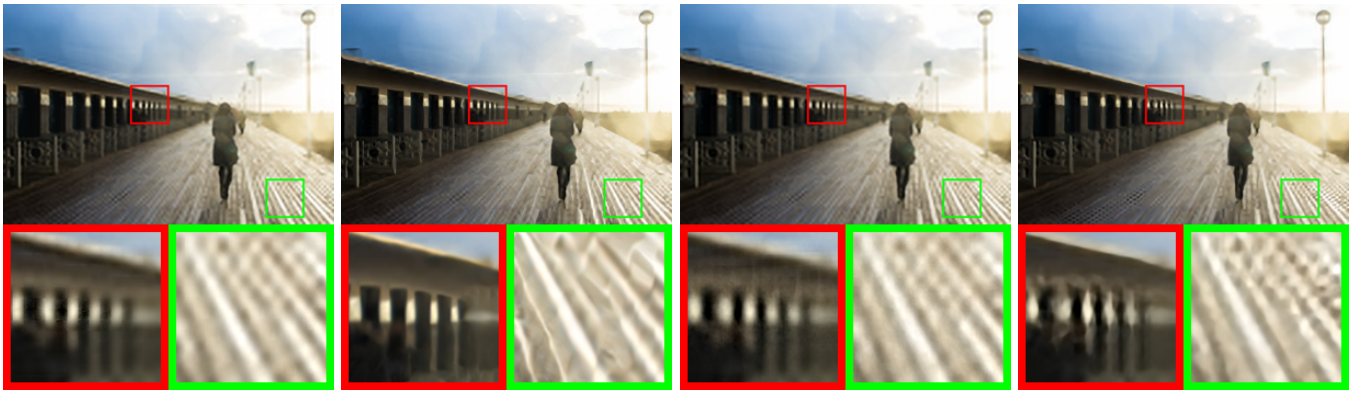


VDSR [8] | LapSRN (ours)

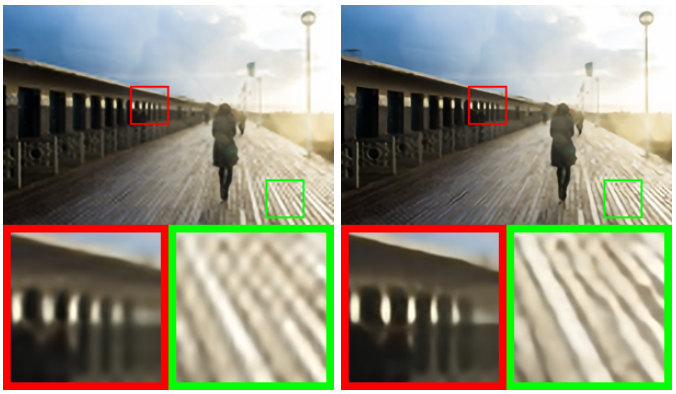
Figure 35: Visual comparison for 8× SR on URBAN100.



HR | Bicubic | A+ [11] | SRCNN [4]



FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]



VDSR [8] | LapSRN (ours)

Figure 36: Visual comparison for 8× SR on URBAN100.

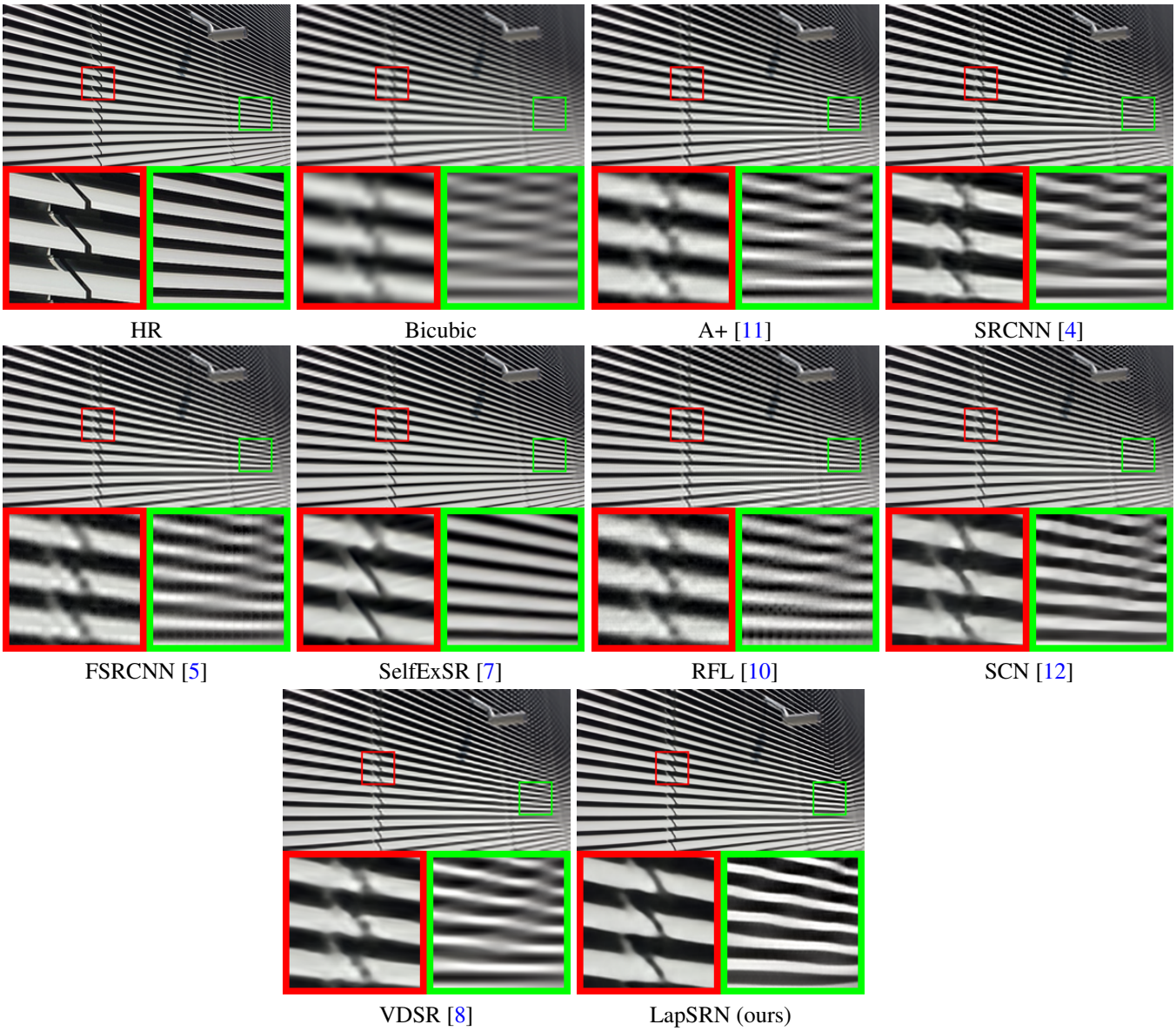
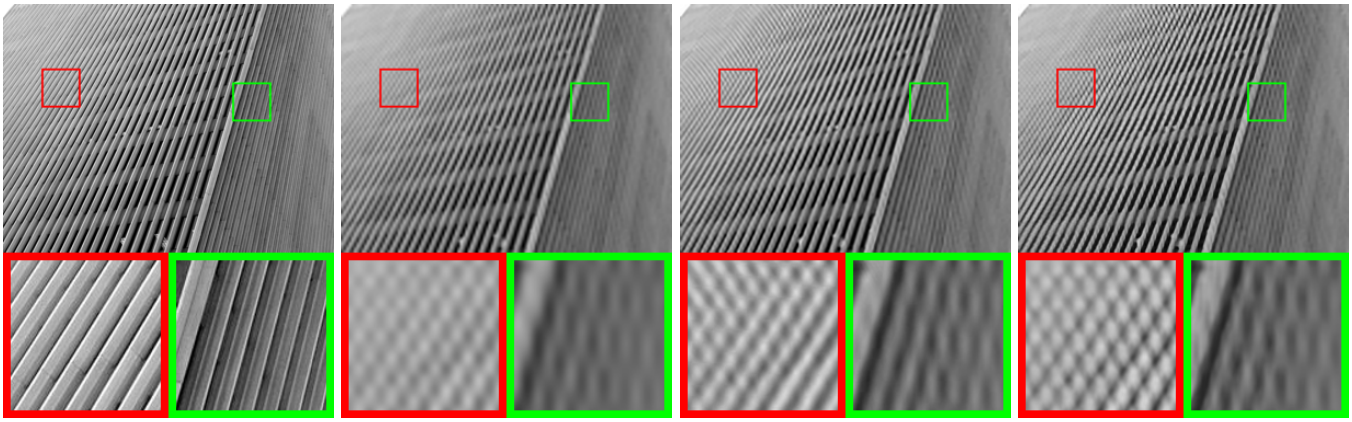
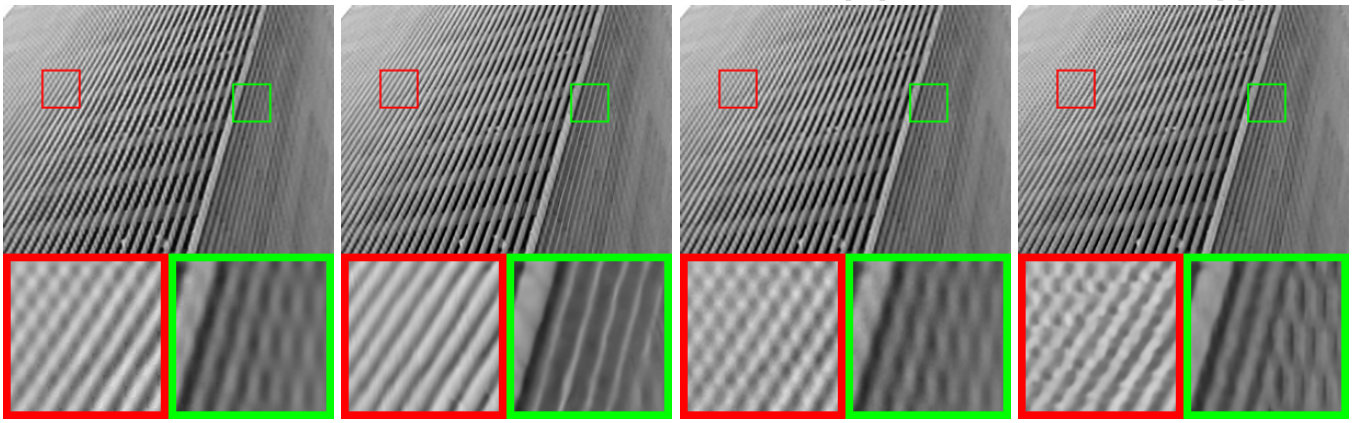


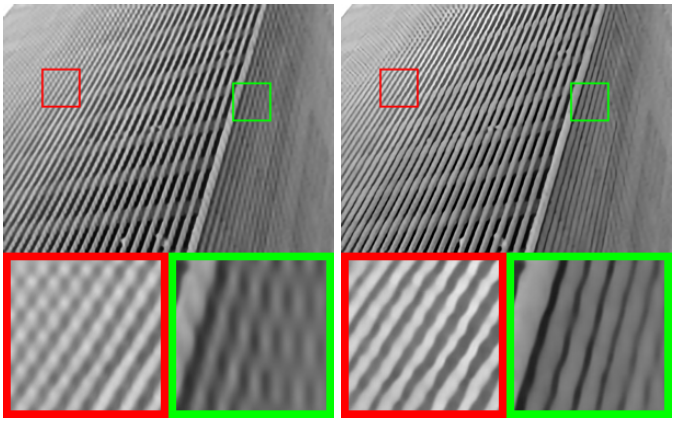
Figure 37: Visual comparison for 8× SR on URBAN100.



HR | Bicubic | A+ [11] | SRCNN [4]

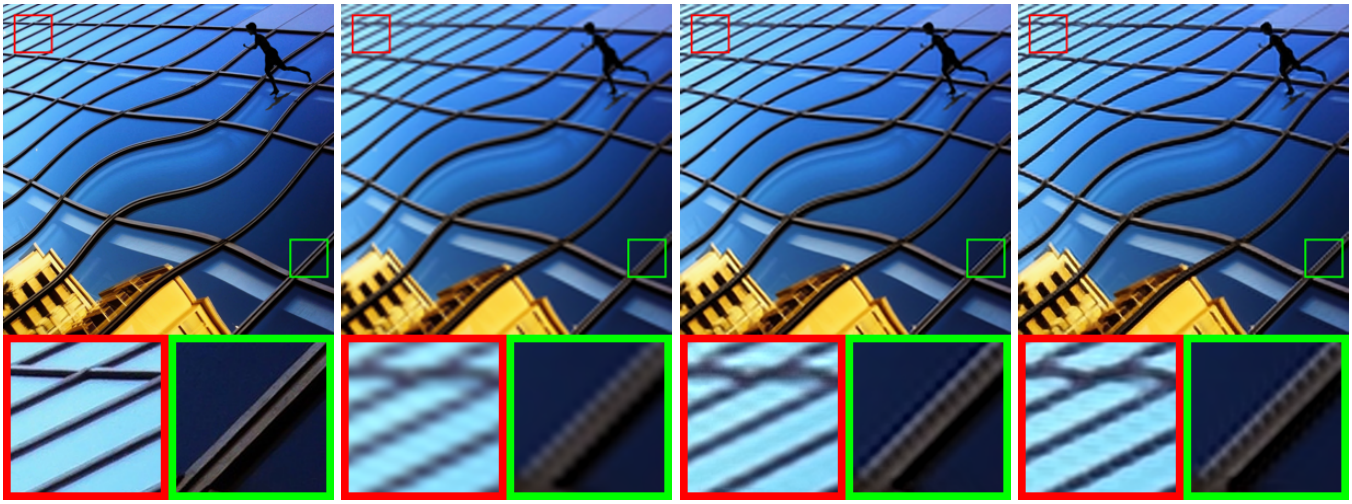


FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

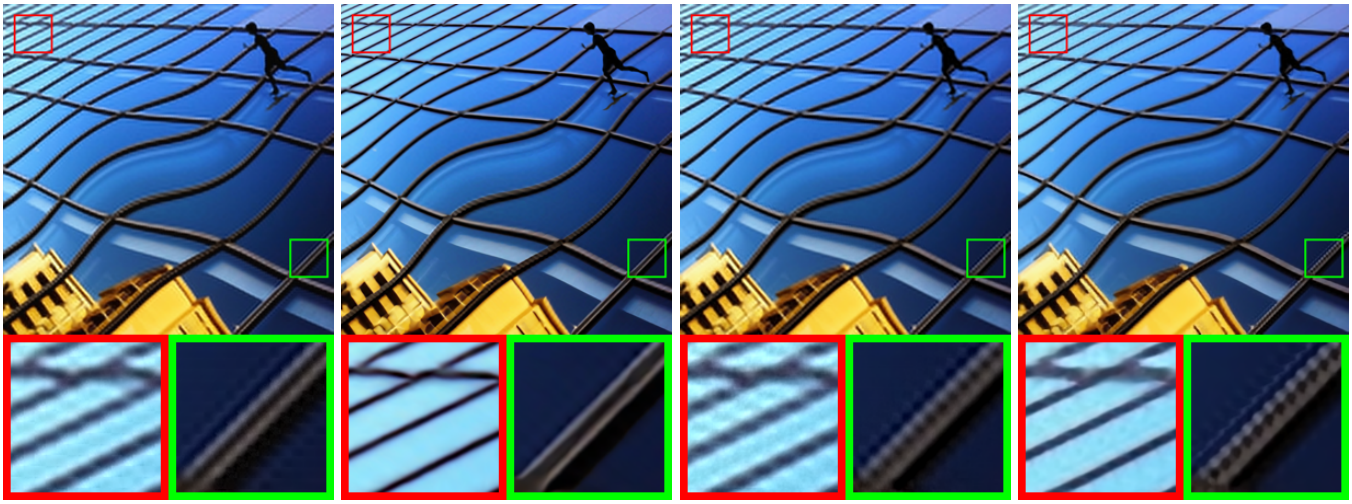


VDSR [8] | LapSRN (ours)

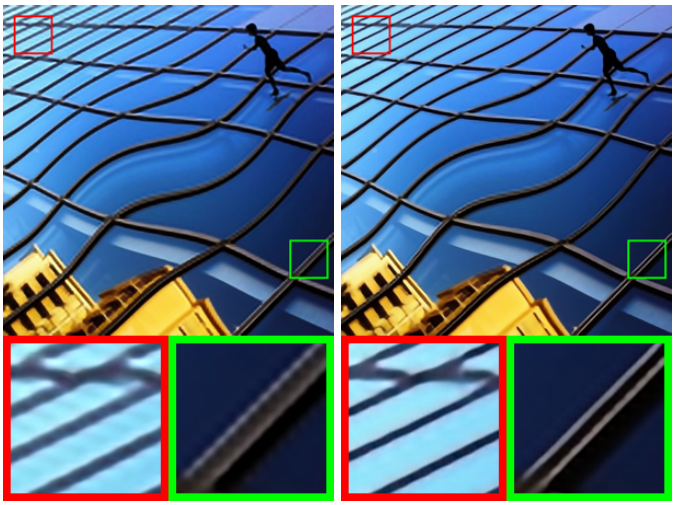
Figure 38: Visual comparison for 8× SR on URBAN100.



HR | Bicubic | A+ [11] | SRCNN [4]

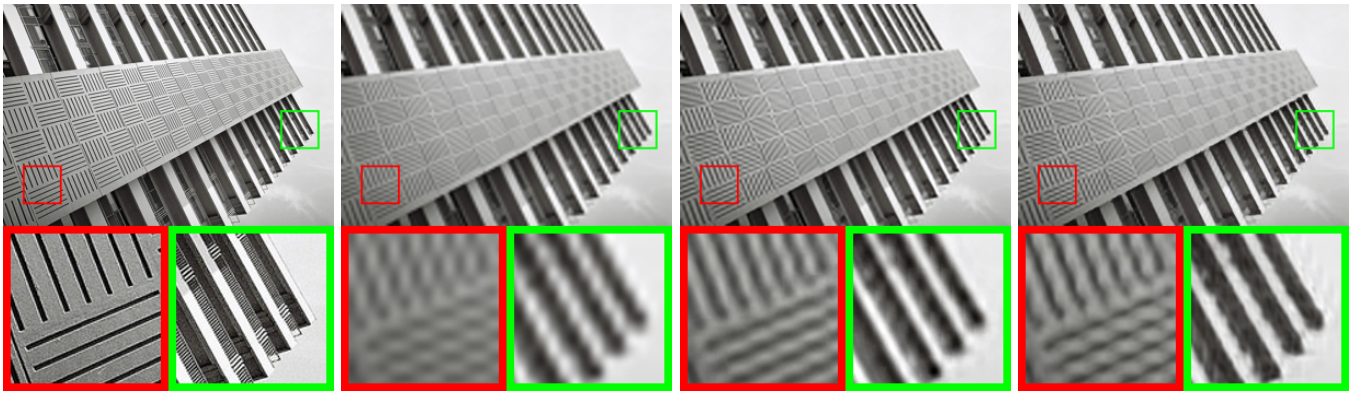


FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

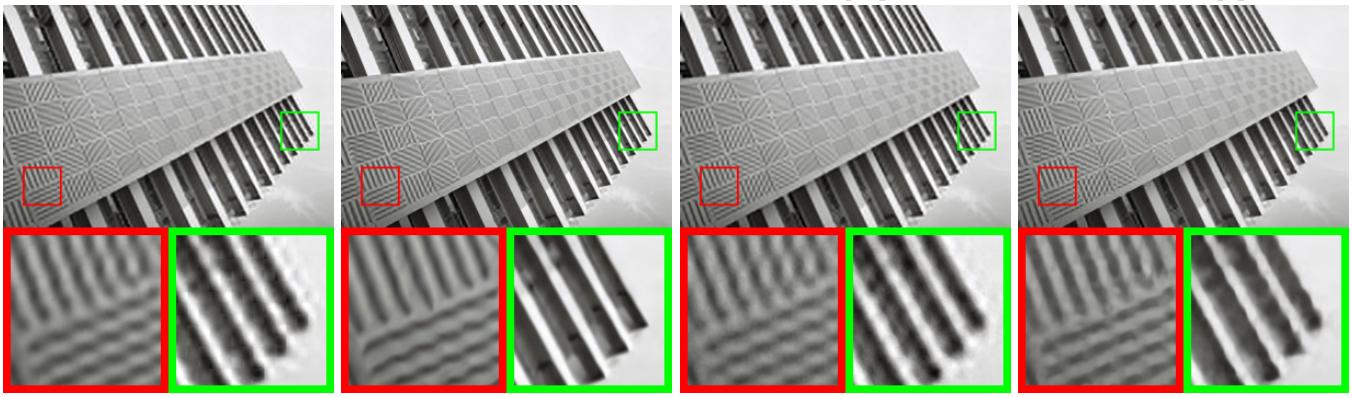


VDSR [8] | LapSRN (ours)

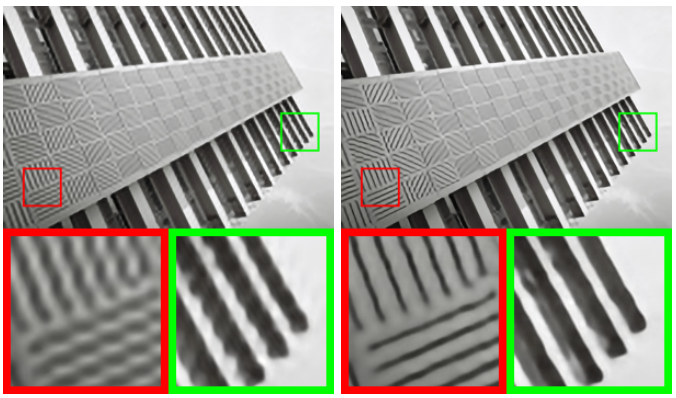
Figure 39: Visual comparison for 8× SR on URBAN100.



HR | Bicubic | A+ [11] | SRCNN [4]



FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]



VDSR [8] | LapSRN (ours)

Figure 40: Visual comparison for 8× SR on URBAN100.

6.6. Visual comparisons on Manga109 with 8× SR

Figure 41: Visual comparison for 8× SR on MANGA109.

HR | Bicubic | A+ [11] | SRCNN [4]

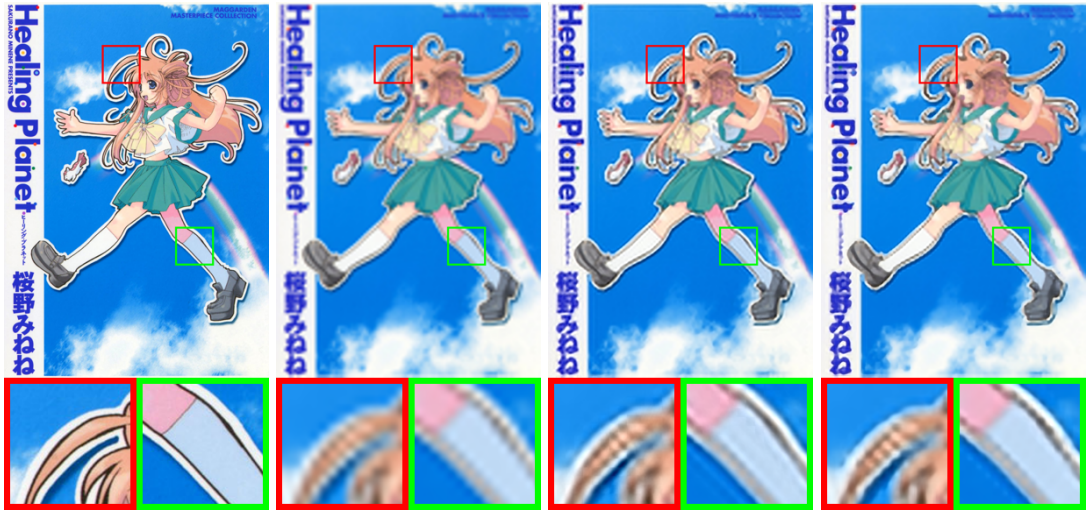
FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]

VDSR [8] | LapSRN (ours)

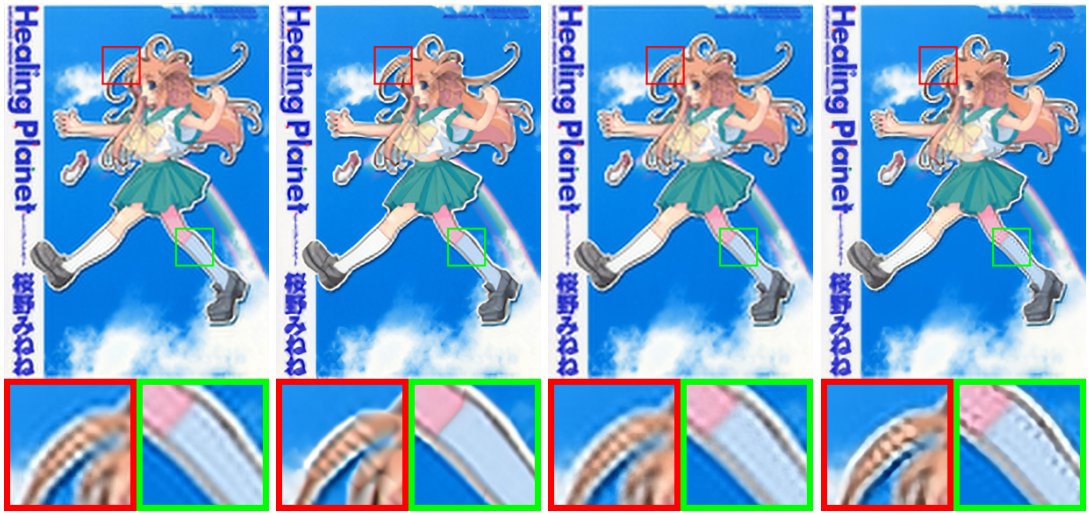
Figure 42: Visual comparison for 8× SR on MANGA109.

Figure 43: Visual comparison for 8× SR on MANGA109.

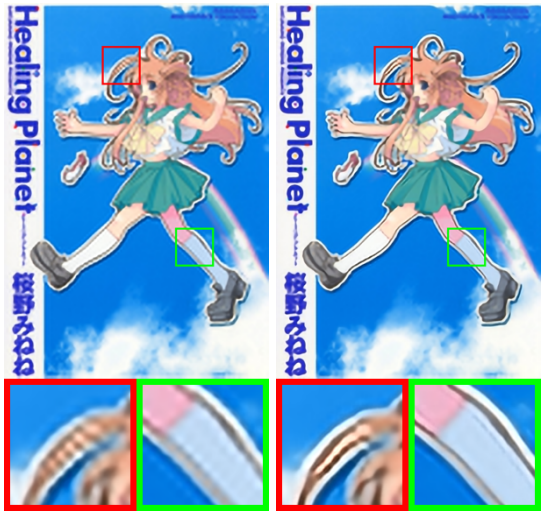
Figure 44: Visual comparison for 8× SR on MANGA109.



HR | Bicubic | A+ [11] | SRCNN [4]



FSRCNN [5] | SelfExSR [7] | RFL [10] | SCN [12]



VDSR [8] | LapSRN (ours)

Figure 45: Visual comparison for 8× SR on MANGA109.

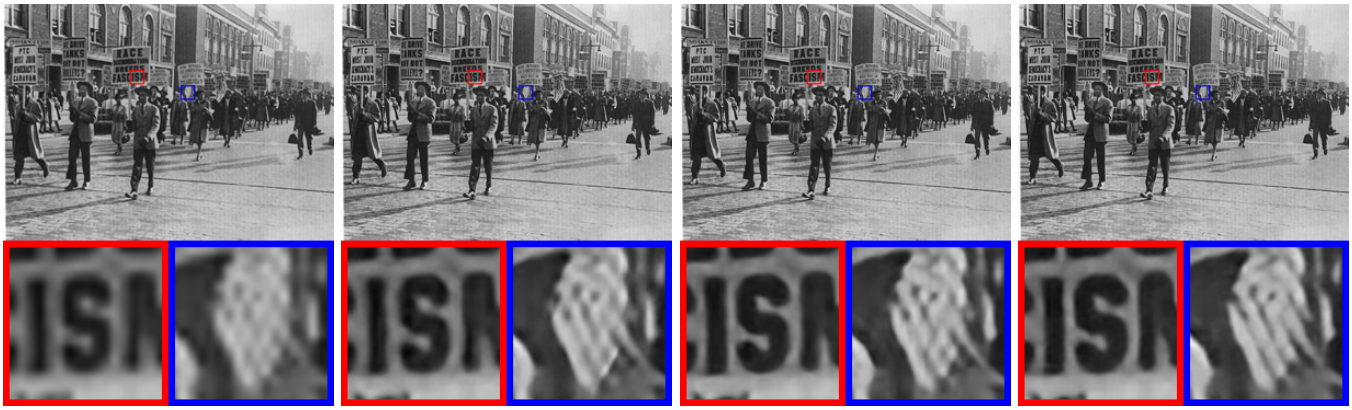
Figure 46: Visual comparison for 8× SR on MANGA109.

Figure 47: Visual comparison for 8× SR on MANGA109.

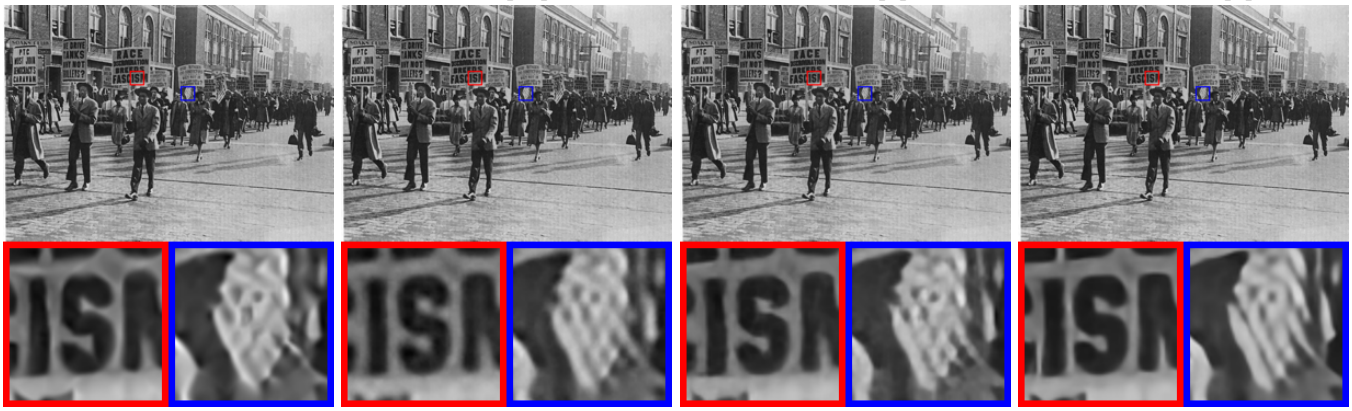
Figure 48: Visual comparison for 8× SR on MANGA109.

Figure 49: Visual comparison for 8× SR on MANGA109.

6.7. Visual comparisons on real-world photos with 4× SR



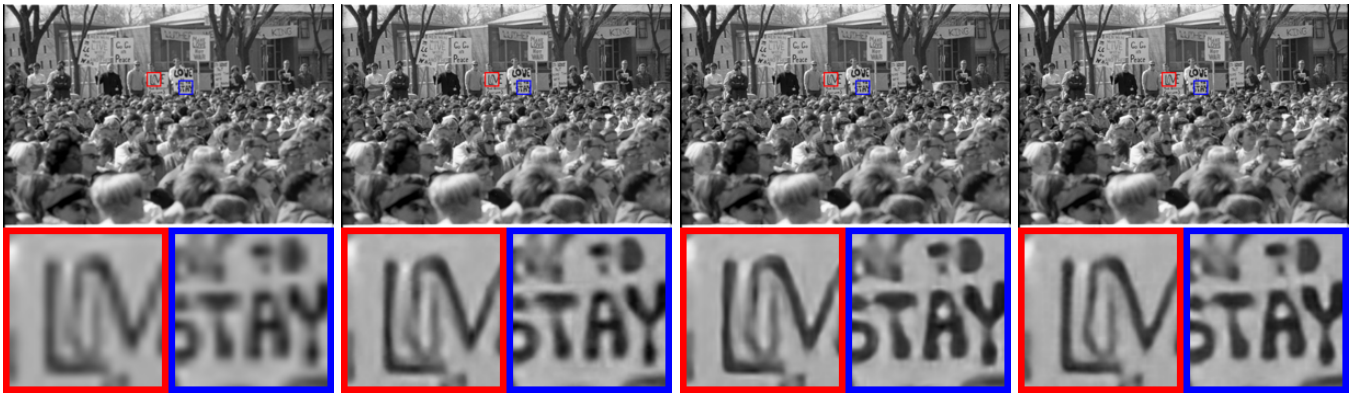
Bicubic | A+ [11] | SRCNN [4] | FSRCNN [5]

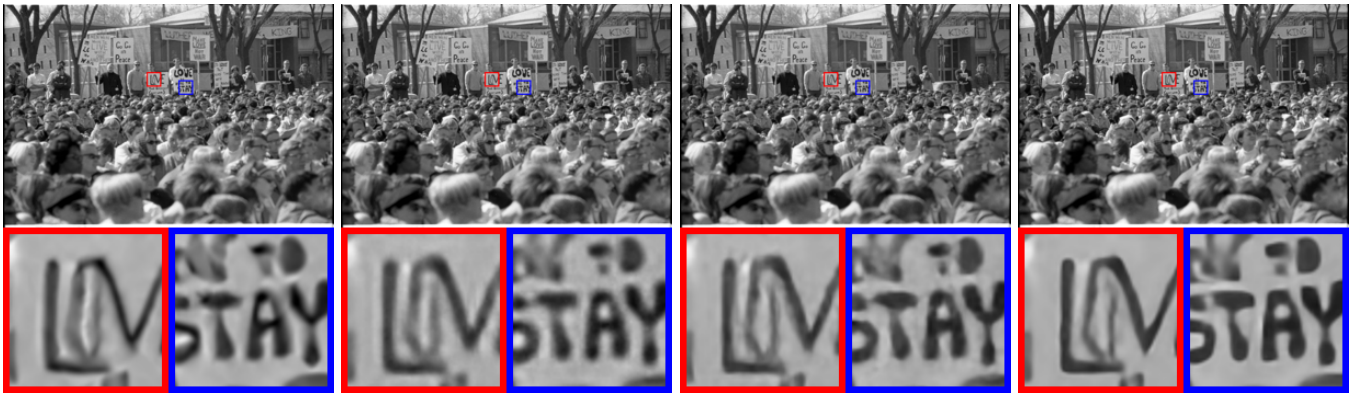


SelfExSR [7] | RFL [10] | SCN [12] | VDSR [8]

DRCN [9] | LapSRN (ours)

Figure 50: Visual comparison for 4× SR on real-world photos.





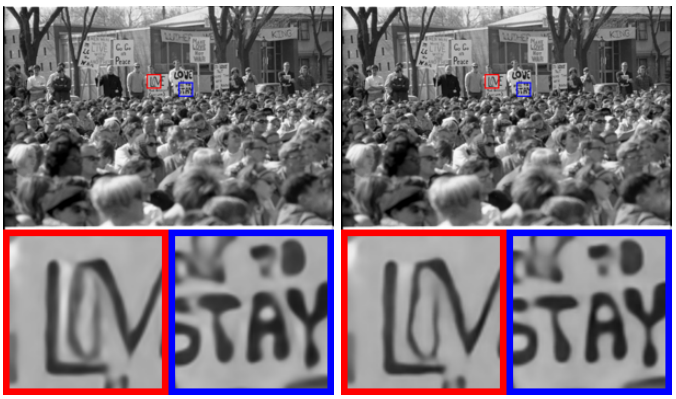


Figure 51: Visual comparison for 4× SR on real-world photos.

Bicubic A+ [11] SRCNN [4] FSRCNN [5]

SelfExSR [7] RFL [10] SCN [12] VDSR [8]

DRCN [9] LapSRN (ours)

Figure 52: Visual comparison for 4× SR on real-world photos.

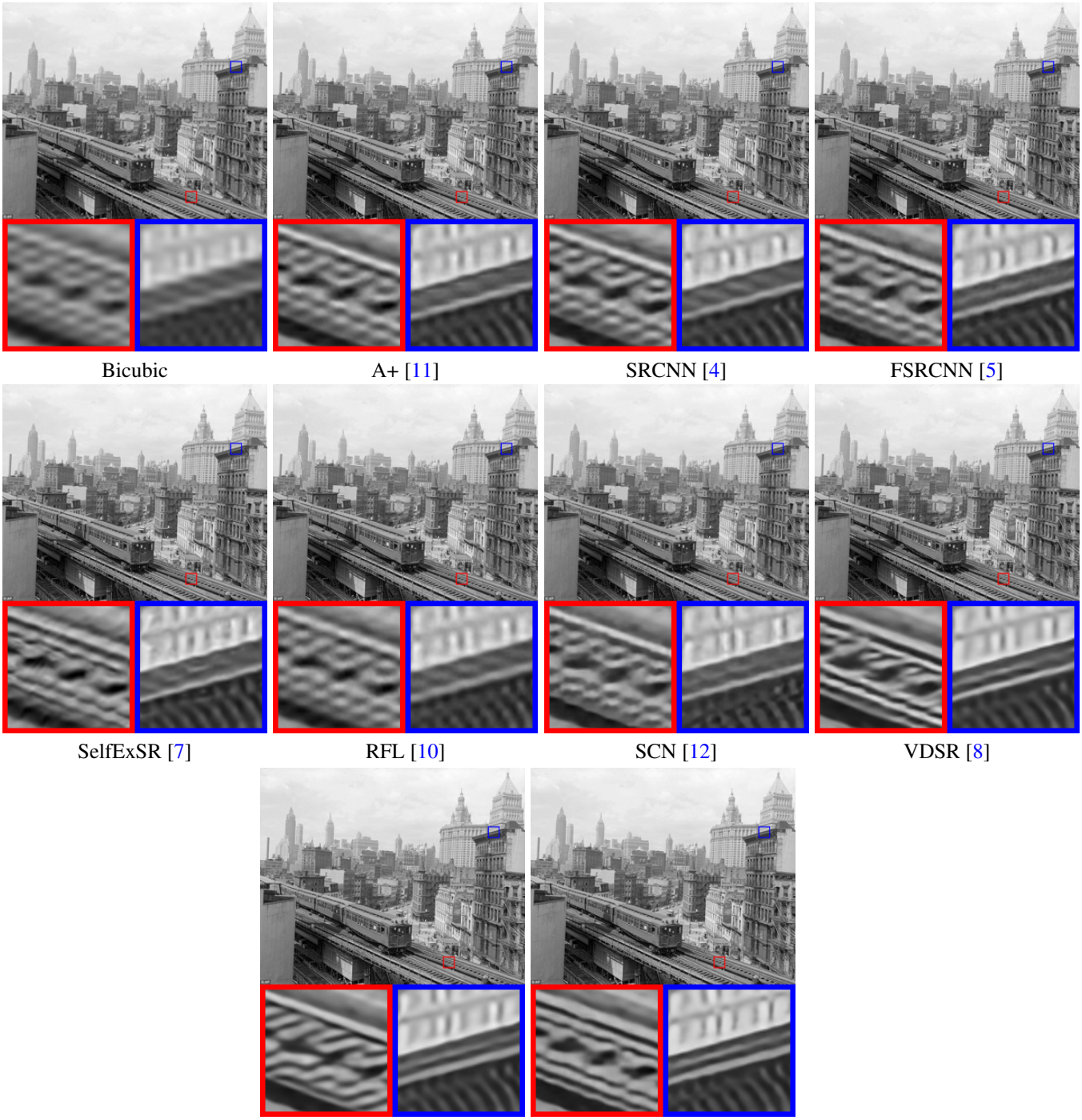


Figure 53: Visual comparison for 4× SR on real-world photos.

Bicubic A+ [11] SRCNN [4] FSRCNN [5]

SelfExSR [7] RFL [10] SCN [12] VDSR [8]

DRCN [9] LapSRN (ours)

Figure 54: Visual comparison for 4× SR on real-world photos.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. 5

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 5

[3] E. L. Denton, S. Chintala, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015. 1, 2, 4

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2015. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56

[5] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 4

[7] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56

[8] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56

[9] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 52, 53, 54, 55, 56

[10] S. Schulter, C. Leistner, and H. Bischof. Fast and accurate image upscaling with super-resolution forests. In *CVPR*, 2015. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56

[11] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56

[12] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *ICCV*, 2015. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56

[13] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *TIP*, 19(11):2861–2873, 2010. 5

[14] X. Yu and F. Porikli. Ultra-resolving face images by discriminative generative networks. In *ECCV*, 2016. 4