CDFormer: When Degradation Prediction Embraces Diffusion Model for Blind Image Super-Resolution

Supplementary Material

6. Preliminaries

Diffusion Models (DMs) rely on a long Markov chain of diffusion steps to generate samples. They first define a forward diffusion process that transforms the input image x_0 to Gaussian noise $x_T \sim \mathcal{N}(0,1)$ over T iterations. Each iteration in the forward process can be described as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{14}$$

where x_t is the noised image at time-step t, β_t is the predefined scale factor. Using a reparameterization trick, the above equation can be simplified as:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{15}$$

where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=0}^{t}\alpha_i$.

The reverse diffusion process is then defined to recreate a sample from $p(x_{t-1} \mid x_t)$ as follows:

$$p(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_t(x_t, x_0), \sigma_t^2\mathbf{I}), \tag{16}$$

where $\boldsymbol{\mu}_t(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}(x_t - \epsilon\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}})$, $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. ϵ, the added noise in the forward process to x_t, however, is unknown in the reverse process. Thus unconditional DMs are trained to predict noise ϵ for each step, denoted as $\epsilon_\theta(x_t, t)$.

Simply, Eq. (16) can be rewritten as:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(x_t, t)) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t, \tag{17}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$ is the added noise at step t.

The training of DMs uses the variational lower bound to optimize the negative log-likelihood:

$$\nabla_{\boldsymbol{\theta}}||\epsilon - \epsilon_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\mathrm{x}_0 + \epsilon\sqrt{1-\bar{\alpha}_t}, t)||_2^2. \tag{18}$$

where θ is the parameters of the network.

7. Algorithm

We provide the training and inference algorithms of $CDFormer_{stage2}$ in Algorithm 1 and Algorithm 2, respectively. Notice that during training $\hat{Z}_T$ is computed from Z_0, which is predicted by E_{GT}, while $\hat{Z}_T$ during inference is sampled from Gaussian noise.

Algorithm 1 $CDFormer_{stage2}$ Training

```
Input: Trained $CDFormer_{stage1}$ (including $E_{GT}$ and $CDFormer_{SR}$), timesteps $T$, schedule $\beta_t, \alpha_t$ ($t \in [1, T]$), $I_{LR}$, $I_{HR}$.
Output: Trained $CDFormer_{stage2}$.

Init: Forzen $E_{GT}$.
for $I_{LR}, I_{HR}$ do
    $Z_0 = E_{GT}(Concat((I_{HR})\downarrow_s, I_{LR}), I_{HR})$.
    Forward Process:
    Sample $Z_T$ by $q(Z_T|Z_0) = \mathcal{N}(Z_T; \sqrt{\bar{\alpha}_T}Z_0, (1-\bar{\alpha}_T)\mathbf{I})$
    Reverse Process:
    $\hat{Z}_T = Z_T$
    $c = E_{LR}(I_{LR})$
    for $t = T$ to 1 do
        $\hat{Z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\hat{Z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\hat{Z}_t, t, c)) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t$
    end for
    $I_{SR} = CDFormer_{SR}(I_{LR}, \hat{Z}_0)$
    Calculate $\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{rec}}$
end for
Output the trained model $CDFormer_{stage2}$.
```

Algorithm 2 $CDFormer_{stage2}$ Inference

```
Input: Trained $CDFormer_{stage2}$ (including $E_{LR}$ and $CDFormer_{SR}$), timesteps $T$, schedule $\beta_t, \alpha_t$ ($t \in [1, T]$), $I_{LR}$.
Output: Reconstructed SR images $I_{SR}$.

Sample $\hat{Z}_T \sim \mathcal{N}(0, 1)$
$c = E_{LR}(I_{LR})$
for $t = T$ to 1 do
    $\hat{Z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\hat{Z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\hat{Z}_t, t, c)) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t$
end for
$I_{SR} = CDFormer_{SR}(I_{LR}, \hat{Z}_0)$
return Reconstruct SR images $I_{SR}$.
```

8. Discussion

Our research has revealed an inherent drawback in the application of diffusion models (either in the pixel space or latent space) for Blind image Super-Resolution (BSR): their generated SR images often exhibit inconsistencies with desired content. We attribute this phenomenon to the designing goal of diffusion models, *i.e.*, DMs are essentially intended for image synthesis rather than image reconstruction. This generative model type leads to an overemphasis on diversity, which we assume to be counterproductive for BSR.

To be specific, when LR images of extremely low quality are input, a scarcity of information can be utilized for reconstruction. In this case, applying diffusion models to reconstruct images will further exacerbate this scarcity, making degradation estimation more difficult and leading to a dominant role of randomness in the reverse process. This explains why traditional deep learning methods can outperform diffusion-based SR approaches in widely used metrics such as PSNR and SSIM. We instead propose a diffusion-based estimator to predict high-level representation. The conditional vector produced by LR images is able to prevent excessive diversity. CDFormer therefore has the ability to achieve a new state-of-the-art performance.

However, our experiments in more complex degradation scenarios, as demonstrated in Tab. 3, revealed only modest performance improvements. We suspect that in situations where degradation reaches a certain level, both traditional deep learning methods and our proposed CDFormer struggle to reconstruct high-resolution images effectively. Therefore, it may be helpful to allow for more diversity in the diffusion process, which we have left as future work.

9. Experiment Settings

All experiments are conducted on GeForce RTX 4090 GPU. The size of the Gaussian kernel is fixed to 21×21. We first train our method on noise-free degradation with isotropic Gaussian kernels only. The ranges of the kernel widths σ are set to $[0.2, 2.0]$, $[0.2, 3.0]$, and $[0.2, 4.0]$ for $\times 2/3/4$ SR, respectively. Then, our method is trained on more general types of degradation with anisotropic Gaussian kernels and noise. Anisotropic Gaussian kernels characterized by a Gaussian probability density function $N(0, \sum)$ (with zero mean and varying covariance matrix $\sum$) are considered. The covariance matrix $\sum$ is determined by two random eigenvalues $\lambda_1, \lambda_2 \sim U(0.2, 4)$ and a random rotation angle $\Theta \sim U(0, \pi)$. The noise level ranges from 0 to 25.

10. Additional Ablation Study

10.1. Effects of Iterations Number.

We performed an ablation study on iteration numbers in our redesigned diffusion model. As plotted in Fig. 9, six settings, $T = \{1, 2, 4, 8, 16, 32\}$ respectively, have been tested. The variance hyperparameters are varied as T changes. The PSNR results indicate that a single iteration step is insufficient to generate a meaningful prior representation, thus limiting the Super-Resolution performance. However, when increasing to 2 steps, CDFormer can reach a fantastic result and the curve actually has converged. This finding verifies that treating the diffusion process as a vector estimator can address the problem of increased time cost in large numbers of iterations T $(1,000$ for example). Meanwhile, the dimension of the latent space $C = 256$ is small. Therefore, the computational complexity is also reasonable.

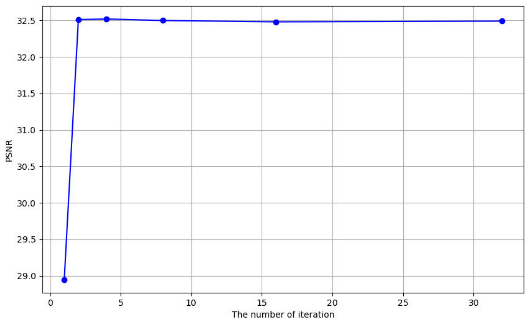


Figure 9. PSNR (↑) results for different settings of iterations number, $T = \{1, 2, 4, 8, 16, 32\}$ respectively.

Effects of Content-aware Degradation-driven Refinement Block (CDRB). We also conduct an ablation study on CDRB to validate the efficiency of $CDFormer_{SR}$ module. As listed in Tab. 6, model5 adopted SwinIR as the SR network, without any proposed module, is inferior to other models. We gradually append the designed module from model6 to model9, resulting in improvements in PSNR and SSIM. Specifically, model6 demonstrates the advantages of integrating spatial attention and channel attention, model9 proves the capacity of CDP and the injection manner.

The Fourier visualization in Fig. 10 further explains how features are modified after depth-wise convolution, self-attention, and fusion. It is obvious that depth-wise convolution focuses on low-frequency information while lacking image edge details. In contrast, self-attention prioritizes high-frequency information but lacks structural information. For example, the sewing of the hat is invisible in Fig. 10a while clear in Fig. 10b, but the nose region represents an opposite phenomenon. By incorporating spatial and channel interactions to fuse two types of feature maps, the results in Fig. 10c exhibit an improved representation with both high- and low-frequency information, indicating the benefits of proposed intra- and inter-path aggregation techniques.

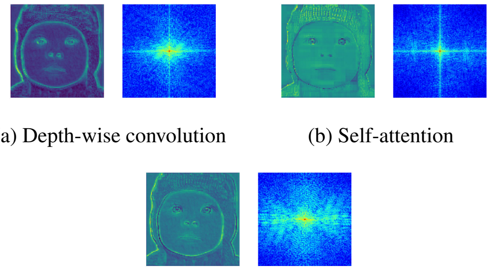
11. More Visualization Results

11.1. Results of Local Attention Map.

To further demonstrate the effectiveness of CDFormer, we utilize integral gradient analysis, LAM , to visualize the pixel influence in Super-Resolution results. As shown in Fig. 11, LAM maps (column 2) exhibit the importance of each pixel in the input LR image w.r.t. the output SR image

Table 6. Ablation study of CDRB on Set5 for different kernel widths are shown. Best in **blod**.

Method	GT CDP	SW-SA	CW-SA	CDIM	0		1.2		2.4		3.6	
					PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
model5	✘	✘	✘	✘	31.843	0.8918	31.936	0.8907	31.692	0.8856	30.074	0.8493
model6	✘	✔	✔	✘	32.114	0.8950	32.074	0.8922	32.006	0.8894	30.712	0.8646
model7	✔	✘	✔	✔	32.344	0.8968	32.432	0.8968	32.268	0.8919	31.042	0.8681
model8	✔	✔	✘	✔	32.413	0.8977	32.515	0.8980	32.335	**0.8928**	31.083	0.8685
model9	✔	✔	✔	✔	**32.485**	**0.8980**	**32.564**	**0.8981**	**32.393**	0.8926	**31.175**	**0.8688**


(a) Depth-wise convolution

(b) Self-attention

(c) Fusion

Figure 10. Visualization results for feature maps and Fourier plots

within the region marked with a red box. Compared to the state-of-the-art degradation prediction (DP) method KDSR, CDFormer presents a stronger relationship on both global and local representations. The proposed CDP and adaptive SR network ensure a better use of LR pixels, and can significantly enhance the quality of reconstruction results. Other quantitative metrics as DI, PSNR, and SSIM, also indicate that our method achieves remarkable superiority.

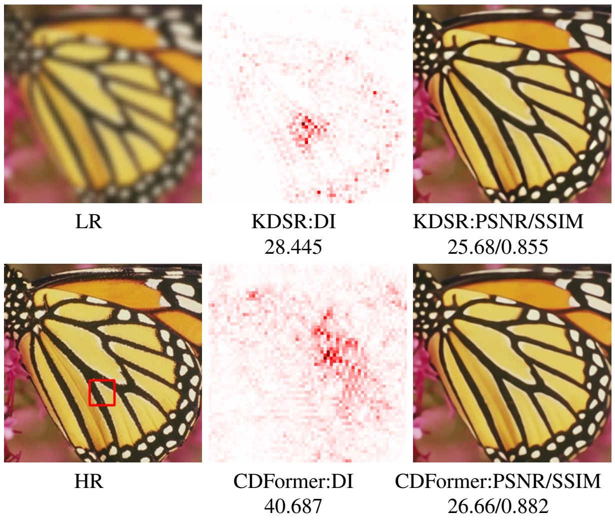
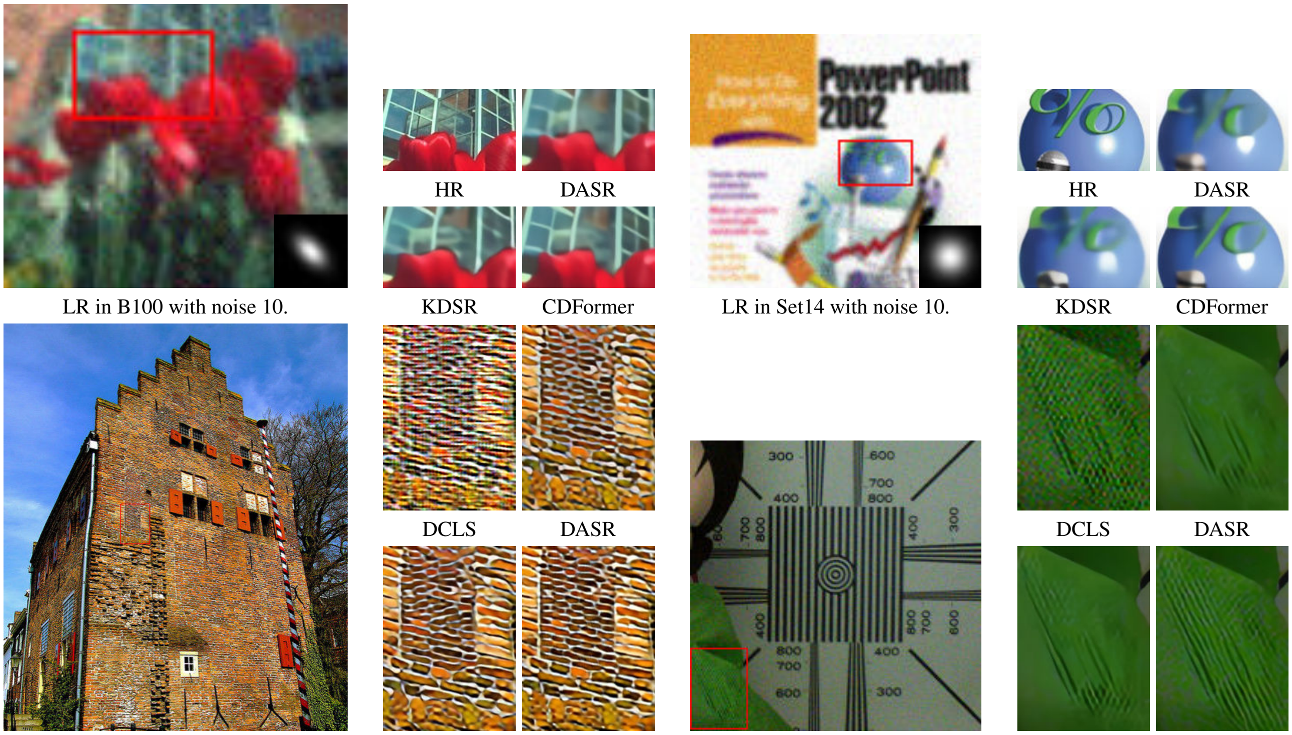


Figure 11. The result of LAM.

11.2. More Visualization Results.

We provide additional visual results in complicated degradation scenarios involving Anisotropic Gaussian Kernels, diverse noises, and real-world conditions in Fig. 12, as well as general degradation scenarios involving Isotropic Gaussian Kernels under noise-free degradation in Figs. 13 to 17.



LR in B100 with noise 10.

HR DASR

KDSR CDFormer

LR in Set14 with noise 10.

HR DASR

KDSR CDFormer

LR in the real-world dataset.

DCLS DASR

KDSR CDFormer

LR in the real-world dataset.

DCLS DASR

KDSR CDFormer

Figure 12. Visualization of different anisotropic Gaussian kernels and noises.

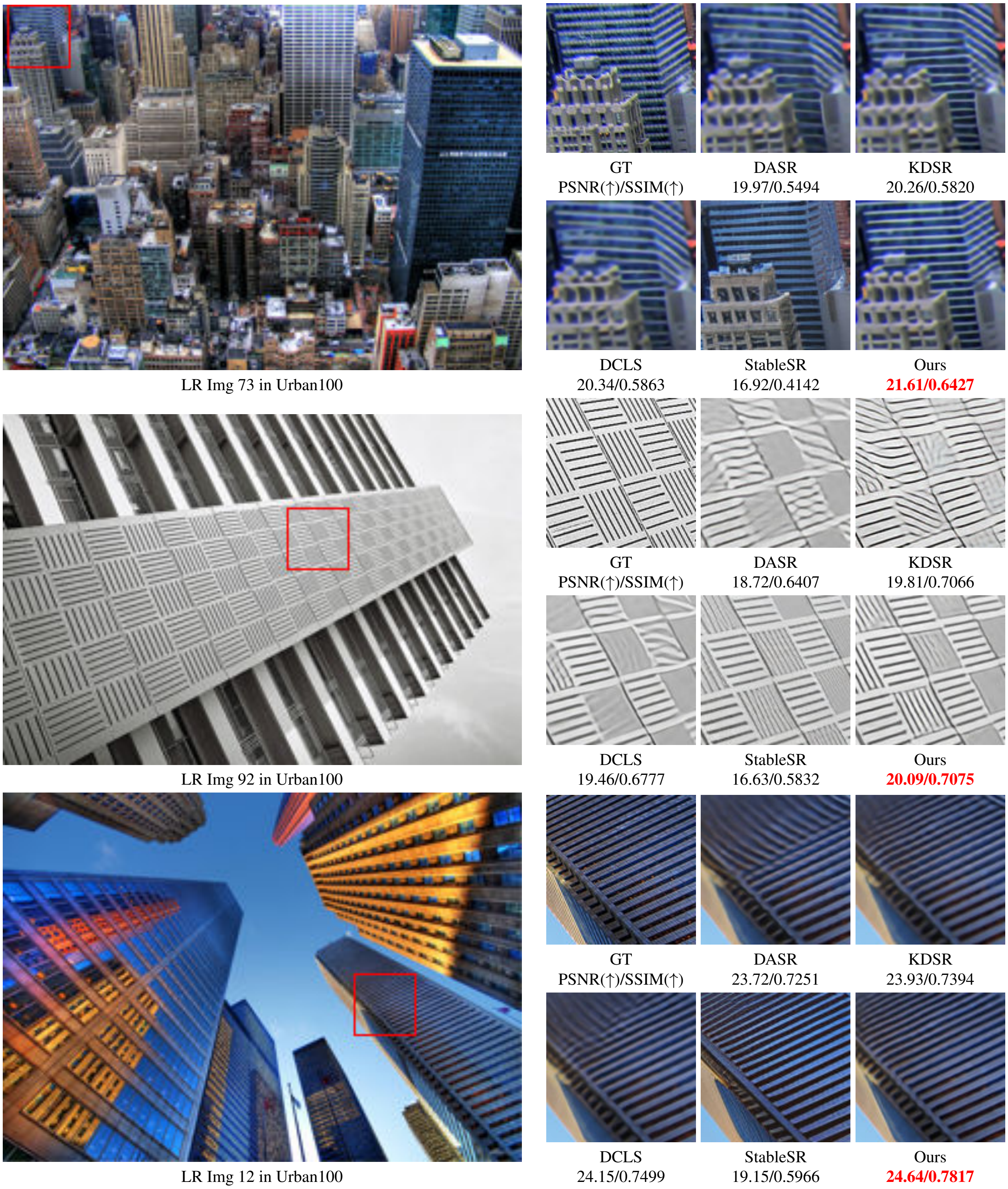
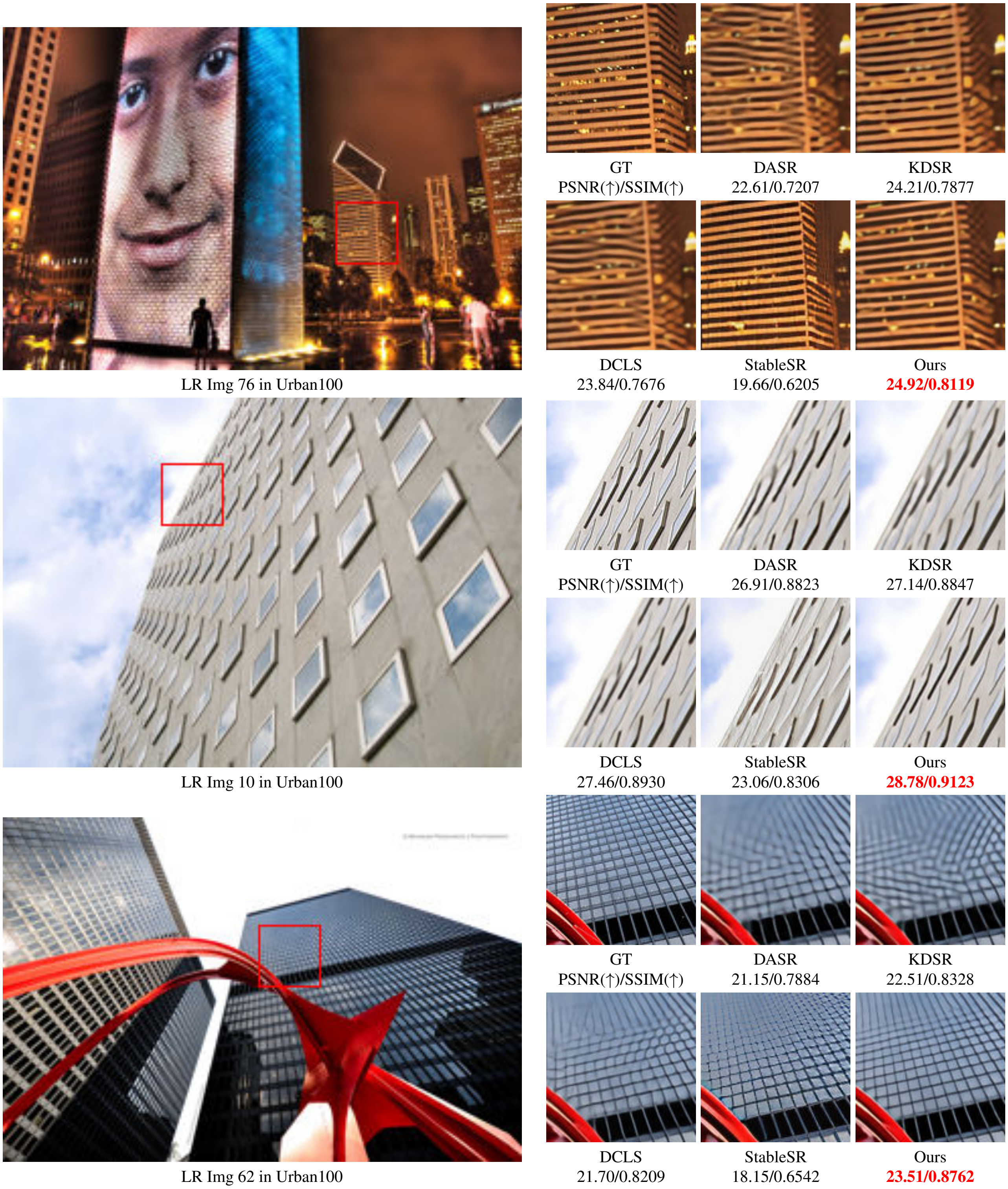


Figure 13. Visual results of Imgs in Urban100, for scale factor 4 and kernel width 0. Best marked in **red**.



LR Img 76 in Urban100

GT PSNR(↑)/SSIM(↑) | DASR 22.61/0.7207 | KDSR 24.21/0.7877

DCLS 23.84/0.7676 | StableSR 19.66/0.6205 | Ours **24.92/0.8119**

LR Img 10 in Urban100

GT PSNR(↑)/SSIM(↑) | DASR 26.91/0.8823 | KDSR 27.14/0.8847

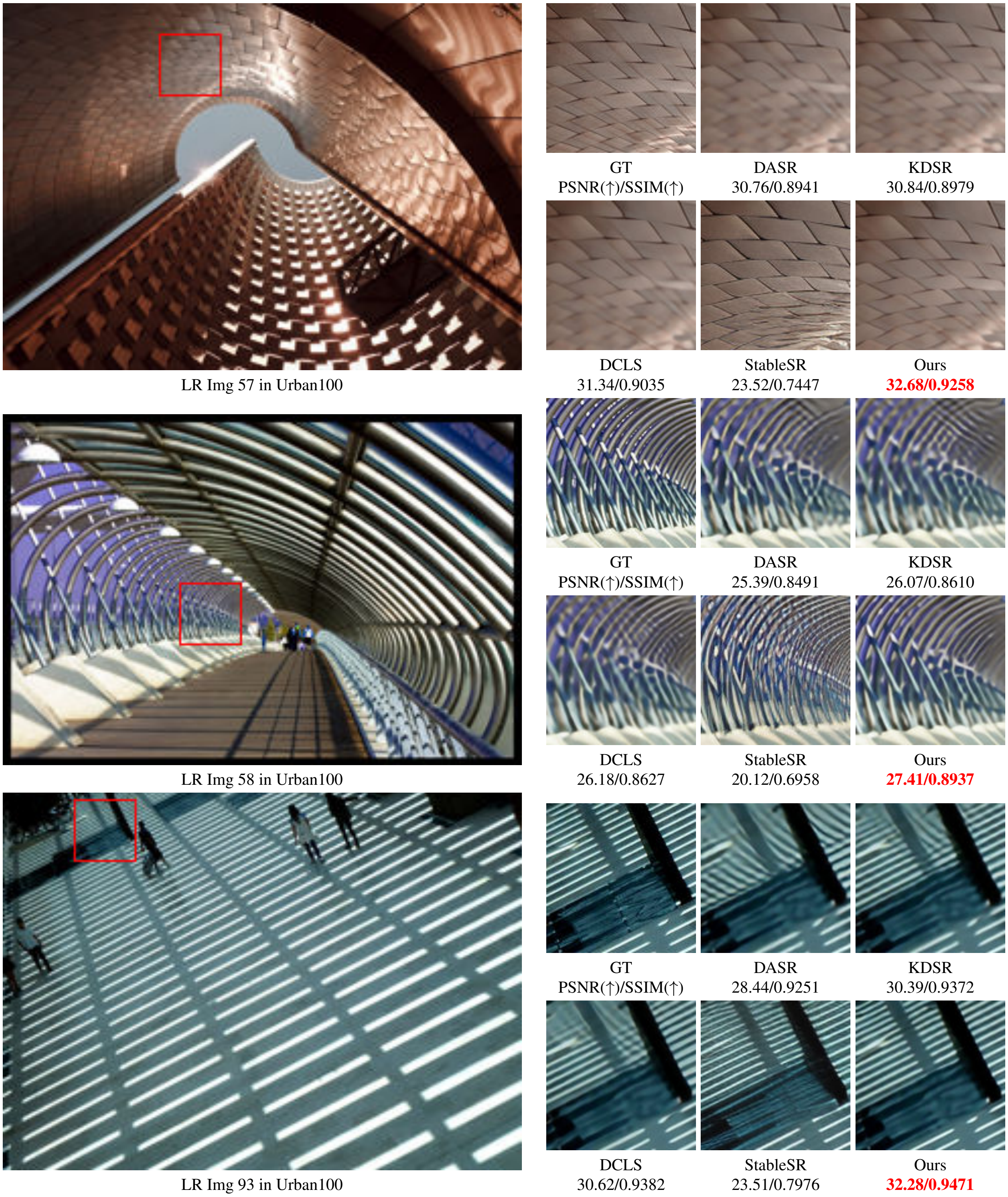
DCLS 27.46/0.8930 | StableSR 23.06/0.8306 | Ours **28.78/0.9123**

LR Img 62 in Urban100

GT PSNR(↑)/SSIM(↑) | DASR 21.15/0.7884 | KDSR 22.51/0.8328

DCLS 21.70/0.8209 | StableSR 18.15/0.6542 | Ours **23.51/0.8762**

Figure 14. Visual results of Imgs in Urban100, for scale factor 4 and kernel width 0. Best marked in **red**.



LR Img 57 in Urban100

GT PSNR(↑)/SSIM(↑) | DASR 30.76/0.8941 | KDSR 30.84/0.8979

DCLS 31.34/0.9035 | StableSR 23.52/0.7447 | Ours **32.68/0.9258**

LR Img 58 in Urban100

GT PSNR(↑)/SSIM(↑) | DASR 25.39/0.8491 | KDSR 26.07/0.8610

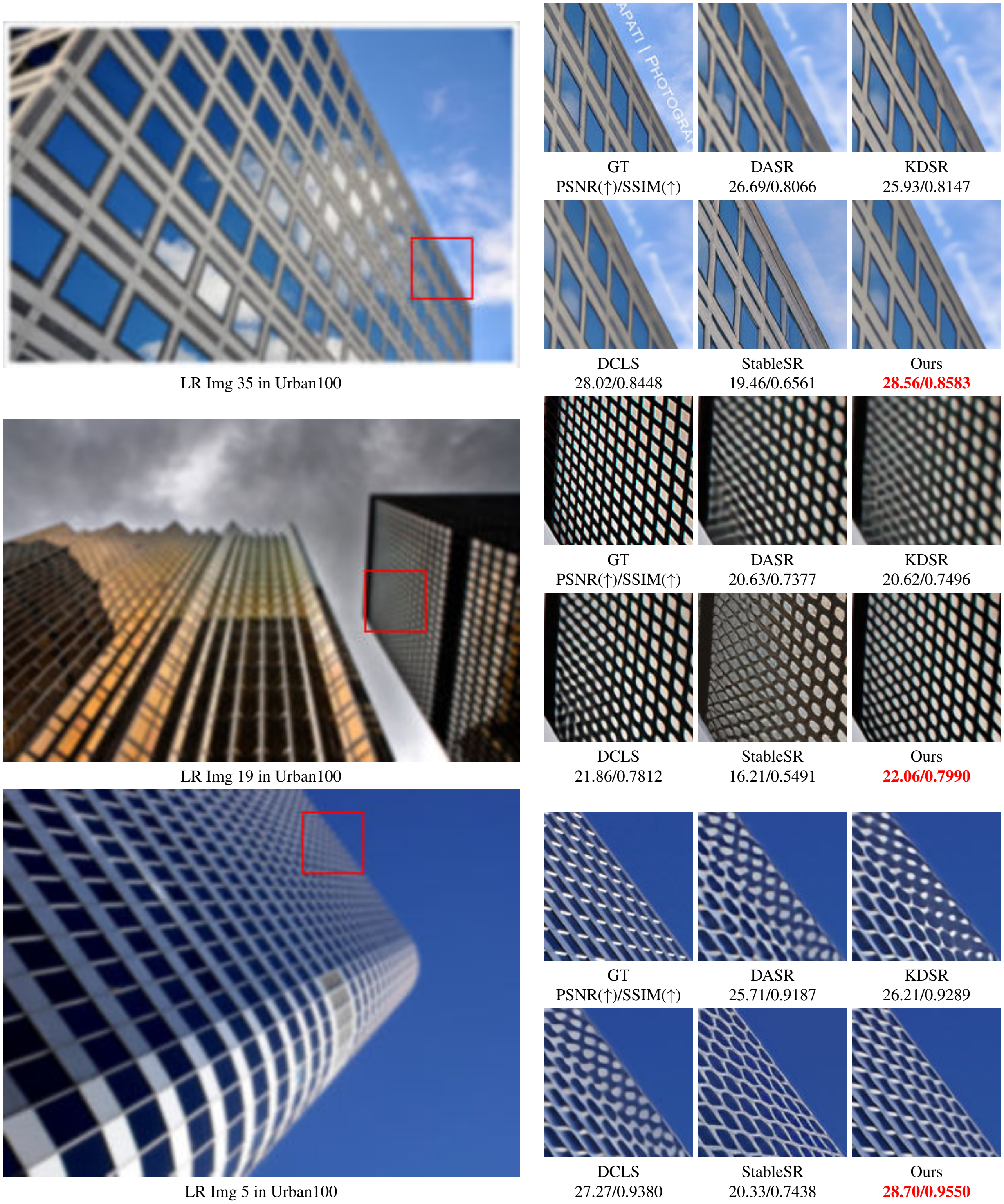
DCLS 26.18/0.8627 | StableSR 20.12/0.6958 | Ours **27.41/0.8937**

LR Img 93 in Urban100

GT PSNR(↑)/SSIM(↑) | DASR 28.44/0.9251 | KDSR 30.39/0.9372

DCLS 30.62/0.9382 | StableSR 23.51/0.7976 | Ours **32.28/0.9471**

Figure 15. Visual results of Imgs in Urban100, for scale factor 4 and kernel width 0. Best marked in **red**.



LR Img 35 in Urban100

GT
PSNR(↑)/SSIM(↑)

DASR
26.69/0.8066

KDSR
25.93/0.8147

DCLS
28.02/0.8448

StableSR
19.46/0.6561

Ours
28.56/0.8583

LR Img 19 in Urban100

GT
PSNR(↑)/SSIM(↑)

DASR
20.63/0.7377

KDSR
20.62/0.7496

DCLS
21.86/0.7812

StableSR
16.21/0.5491

Ours
22.06/0.7990

LR Img 5 in Urban100

GT
PSNR(↑)/SSIM(↑)

DASR
25.71/0.9187

KDSR
26.21/0.9289

DCLS
27.27/0.9380

StableSR
20.33/0.7438

Ours
28.70/0.9550

Figure 16. Visual results of Imgs in Urban100, for scale factor 4 and kernel width 3.6. Best marked in **red**.

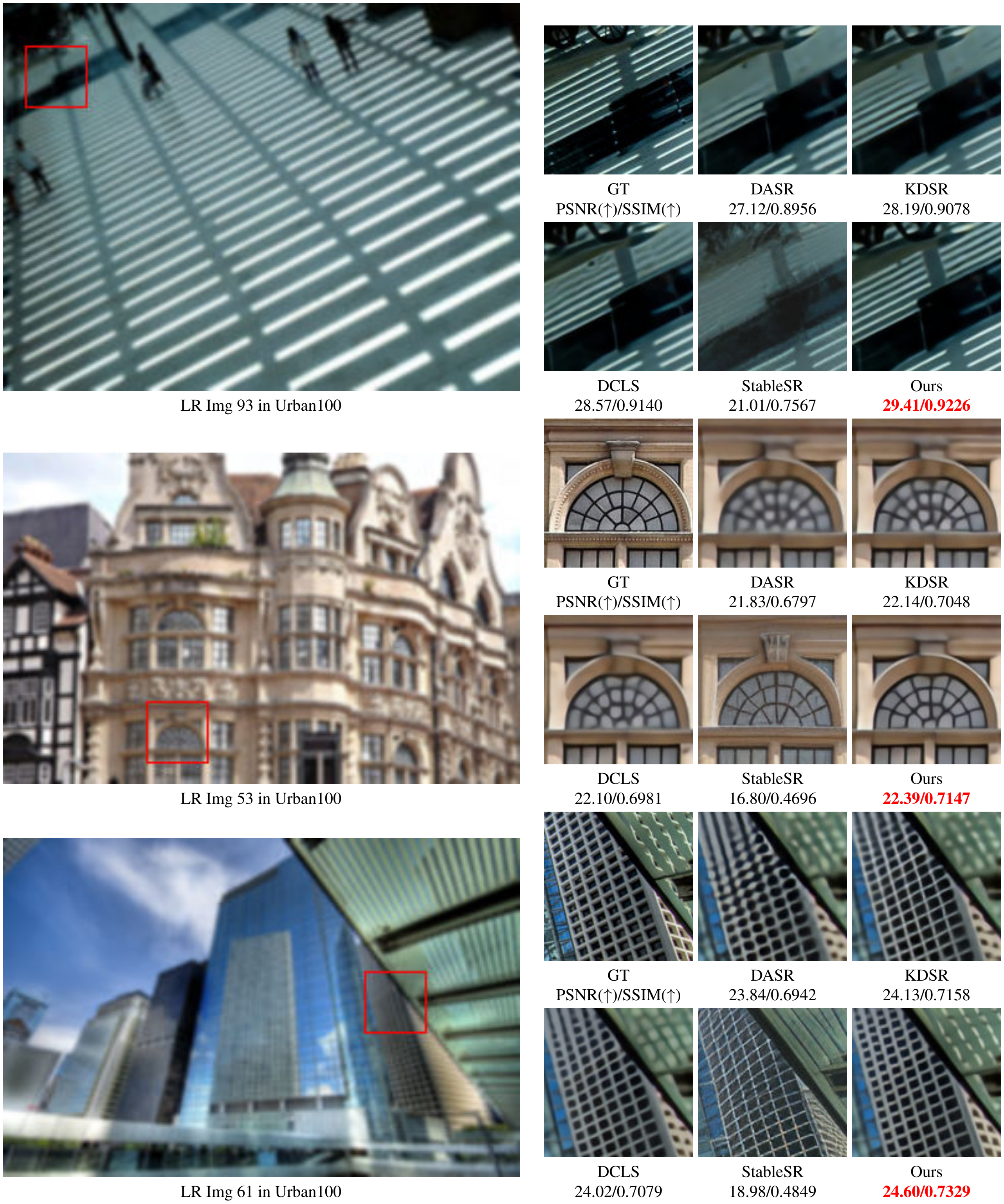


Figure 17. Visual results of Imgs in Urban100, for scale factor 4 and kernel width 3.6. Best marked in **red**.