Consistency Trajectory Matching for One-Step Generative Super-Resolution

Supplementary Material

In the supplementary materials, we introduce more details of our implementation, more experimental results and more visual comparisons.

A. Implementation Details

A.1. Noise and Residual Schedules

Following [12], we design the schedule for $\sigma(t)$ as follows:

$$\sigma(t) = \sigma_{\text{max}} \cdot \left(\frac{t}{T}\right)^{\rho_n}, \tag{22}$$

where σ_{max} denotes the highest noise level and ρ_n controls the speed of noise growth; a larger ρ_n leads to faster growth in the earlier stages and slower growth in the later stages, and vice versa. Similarly, we also design a schedule for $\alpha(t)$:

$$\alpha(t) = \left(\frac{t}{T}\right)^{\rho_r}, \tag{23}$$

where ρ_r serves a role identical to that of ρ_n. In practice, we adopt the linear schedule by setting $\rho_n = 1$ and $\rho_r = 1$.

A.2. Step Schedule

We design a step schedule for Consistency Training of our SR model that adjusts the number of steps with the growth of training iterations. In contrast to [28, 30], we utilize a linearly decreasing curriculum for the total steps T, rather than an increasing one. Specifically, the curriculum is formulated as follows:

$$T(k) = \max(s_0 - \lfloor \frac{k}{K'} \rfloor, s_1), \quad K' = \left\lfloor \frac{K}{s_0 - s_1 + 1} \right\rfloor, \tag{24}$$

where k denotes the training iteration, s_0 denotes the initial steps, s_1 denotes the final steps and K denotes the total iterations. We empirically discover that the decreasing step schedule could produce better results and achieve faster convergence with $s_0 = 4, s_1 = 3$.

A.3. Training Details of Distribution Trajectory Matching

To stabilize the training of DTM, we propose to periodically update $f_{\theta'}$. Specifically, we update $f_{\theta'}$ with the parameters of f_θ every 1k iterations during the training stage of DTM. Algorithm 3 shows the details of the overall training process of CTMSR and Algorithm 3 shows the implementation of Distribution Trajectory Matching loss.

Algorithm 2 Overall training procedure of CTMSR.

```
Require: training CTMSR $f_\theta(\cdot)$
Require: Paired training dataset $(X, Y)$
 1: Stage 1: Consistency Training for One-Step SR
 2: $k \leftarrow 0$
 3: while not converged do
 4:     $\theta^- \leftarrow$ stopgrad($\theta$)
 5:     sample $x_0, y_0 \sim (X, Y)$
 6:     sample $t \sim U(0, T(k) - 1)$
 7:     compute $x_t, x_{t-1}$ using Eq. 1
 8:     $\mathcal{L}_{\text{CT}} = d(f_\theta(x_t, y_0, t), f_{\theta^-}(x_{t-1}, y_0, t-1))$
 9:     Take a gradient descent step on $\nabla_\theta \mathcal{L}_{\text{CT}}$
10:     $k \leftarrow k + 1$
11: end while
12: Stage 2: Distribution Trajectory Matching
13: $\theta' \leftarrow$ stopgrad($\theta$)
14: while not converged do
15:     if $k \equiv 0 \pmod{1000}$ then
16:         $f_{\theta'} \leftarrow f_\theta$
17:     end if
18:     sample $x_0, y_0 \sim (X, Y)$
19:     sample $t' \sim U(1, T(k))$
20:     compute $x_{t'}$ using Eq. 1
21:     $\hat{x}_0 = f_\theta(x_{t'}, y_0, t')$
22:     sample $t \sim U(T_{\text{min}}, T_{\text{max}})$
23:     compute $x_t, \hat{x}_t$ using Eq. 1
24:     grad $= \omega(t)(f_{\theta'}(\hat{x}_t, y_0, t) - f_{\theta'}(x_t, y_0, t)$
25:     $\mathcal{L}_{\text{DTM}} = 0.5 *$ LPIPS($\hat{x}_0$, stopgrad($\hat{x}_0 -$ grad))
26:     $\mathcal{L}_{\text{total}} = \lambda_{\text{CT}}\mathcal{L}_{\text{CT}} + \lambda_{\text{DTM}}\mathcal{L}_{\text{DTM}}$
27:     Take a gradient descent step on $\nabla_\theta \mathcal{L}_{\text{total}}$
28:     $k \leftarrow k + 1$
29: end while
30: return Converged CTMSR $f_\theta(\cdot)$.
```

Algorithm 3 Distribution Trajectory Matching Loss.

```
Require: pre-trained CTMSR $f_{\theta'}(\cdot)$, HR image $x_0$, LR
    image $y_0$, timestep intervals ($T_{\text{min}}, T_{\text{max}}$), SR output $\hat{x}_0$
1: sample $t \sim U(T_{\text{min}}, T_{\text{max}})$
2: compute $x_t, \hat{x}_t, \omega(t)$
3: grad $= \omega(t)(f_{\theta'}(\hat{x}_t, y_0, t) - f_{\theta'}(x_t, y_0, t))$
4: $\mathcal{L}_{\text{DTM}} = 0.5 *$ LPIPS($\hat{x}_0$, stopgrad($\hat{x}_0 -$ grad))
5: return $\mathcal{L}_{\text{DTM}}$
```

A.4. Overall Training Process

The training process of our CTMSR can be broadly divided into two stages as mentioned in the main paper. In the first stage, we train our model exclusively with $\mathcal{L}_{\text{CT}}$ until convergence. Then we utilize a weighted combination of $\mathcal{L}_{\text{CT}}$ and $\mathcal{L}_{\text{DTM}}$ to further optimize our model. The total loss is formulated as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{CT}} \mathcal{L}_{\text{CT}} + \lambda_{\text{DTM}} \mathcal{L}_{\text{DTM}}, \tag{25}$$

where we assign $\lambda_{\text{CT}} = 1$ and $\lambda_{\text{DTM}} = 1.6$. The overall training process is summarized in Algorithm 2.

B. More Experimental Results

B.1. Ablation Study

To comprehensively demonstrate the effectiveness of the proposed DTM, we present additional experimental results of the ablation study on *ImageNet-Test*, RealSet65 and RealSR datasets. The results demonstrate the effectiveness of DTM across synthetic and real-world datasets. The detailed results are shown in Table 5, 6, 7.

B.2. Compared with SinSR

The test results on RealSet65 and RealSR (shown in Table 2) demonstrate that our method outperforms SinSR [36] across all metrics except for CLIPIQA. Upon detailed observation, we discover that the CLIPIQA tends to favor images with noise or artifacts and sometimes fails to distinguish between fine image details and noise or artifacts. Therefore, CLIPIQA occasionally produces higher scores for images of lower quality due to the presence of noise or artifacts. The visual examples are shown in Figure 6.

B.3. Compared with Stable Diffusion-Based Methods

Though Stable Diffusion-based methods achieve impressive results, they rely on the powerful generative capabilities of Stable Diffusion (SD). This results in these methods being constrained by fixed backbones (Stable Diffusion), which limits their scalability to smaller models and consequently restricts their applicability in practical scenarios. In addition, these methods require extremely large models and incur significant inference costs, placing them in a different track from our approach. To compare with SD-based methods, we apply our approach to the latent space provided by VQ-VAE to further enhance the performance of our model. As shown in Table 8, our refined method attains performance on par with SD-based methods with much fewer model parameters and inference time. To be more specific, (1) OSEDiff demands **1.7** times the inference time and **8** times the number of model parameters; (2) AddSR demands **3.7** times the inference time and **10** times the number of model parameters.

Methods	PSNR↑	LPIPS↓	CLIPIQA↑	MUSIQ↑
CTMSR (w/o DTM)	24.71	0.2004	0.6092	56.650
CTMSR (w/ SDS)	23.17	0.2545	0.6292	58.188
CTMSR (w/ DTM)	**24.73**	**0.1969**	**0.6913**	**60.142**

Table 5. Experimental results of ablation study on *ImageNet-Test*.

Methods	CLIPIQA↑	MUSIQ↑	MANIQA↑	NIQE↓
CTMSR (w/o DTM)	0.6009	64.274	0.3658	**4.37**
CTMSR (w/ SDS)	0.6446	62.217	0.3606	4.77
CTMSR (w/ DTM)	**0.6893**	**67.173**	**0.4360**	4.51

Table 6. Experimental results of ablation study on RealSet65.

Methods	CLIPIQA↑	MUSIQ↑	MANIQA↑	NIQE↓
CTMSR (w/o DTM)	0.5542	62.351	0.3512	**4.33**
CTMSR (w/ SDS)	0.6101	60.919	0.3479	5.11
CTMSR (w/ DTM)	**0.6449**	**64.796**	**0.4157**	4.65

Table 7. Experimental results of ablation study on RealSR.

Methods	Runtime (s)	Params (M)	CLIPIQA↑	MUSIQ↑	MANIQA↑
OSEDiff	0.3100	1775	0.6693	**69.10**	0.4717
AddSR	0.6857	2280	0.5410	63.01	0.4113
CTMSR	**0.1847**	**225**	**0.7420**	64.81	**0.4810**

Table 8. Quantitative comparisons with SD-based methods on RealSR. The runtime is tested on 128×128 input images.

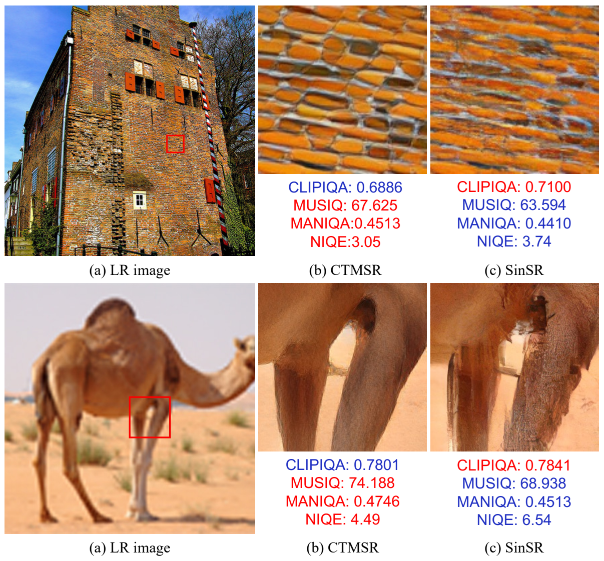


Figure 6. An illustration of CLIPIQA's tendency to favor images with noise or artifacts and its inability to effectively distinguish between fine image details and noise or artifacts. Here are two visual examples of CTMSR and SinSR.

B.3. Visual Comparison

We provide more visual examples of CTMSR compared with recent state-of-the-art methods on *ImageNet-Test* and real-world datasets. The visual examples are shown in Figure 7, 8, 9, 10 11, 12, 13.

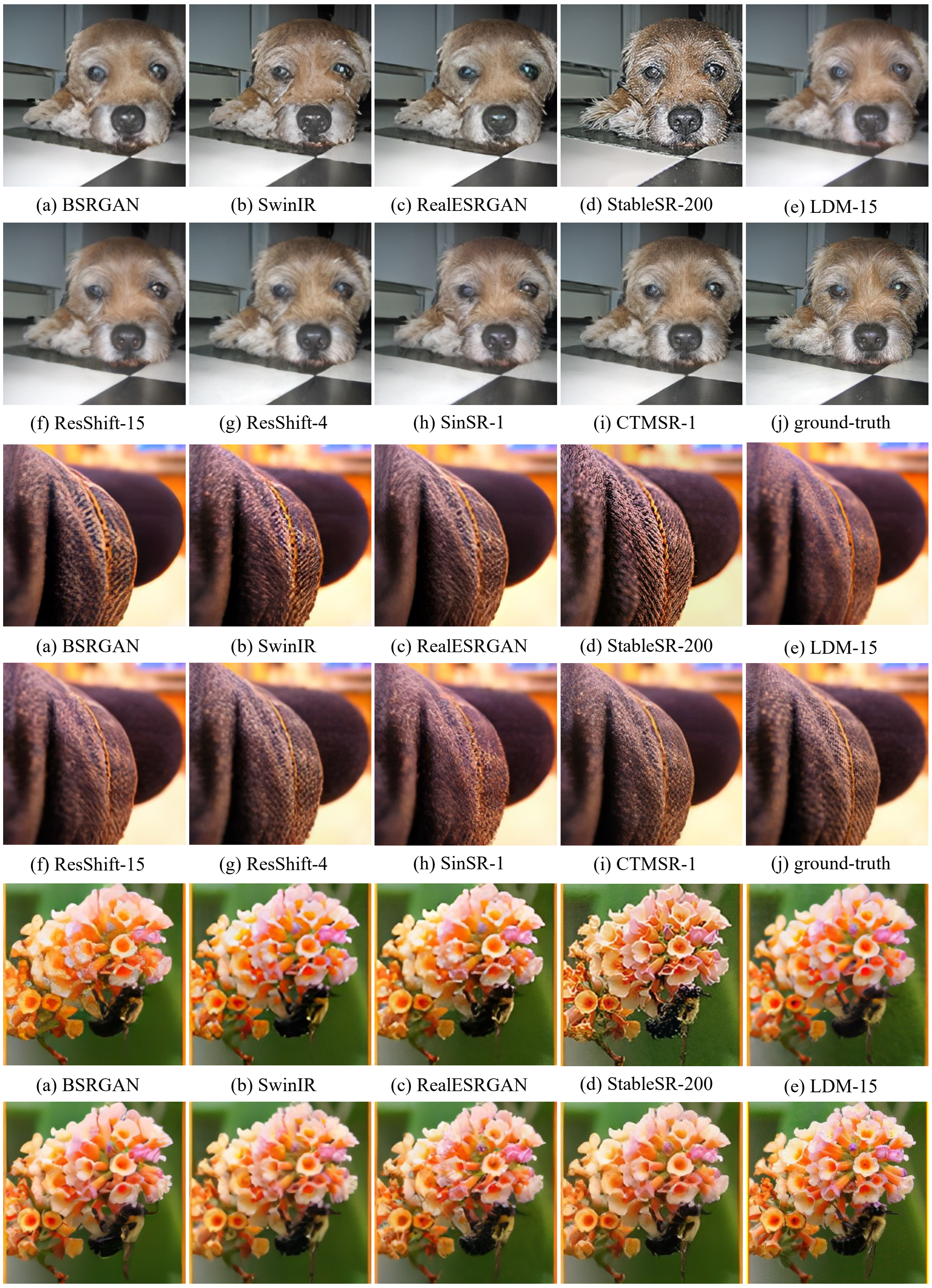
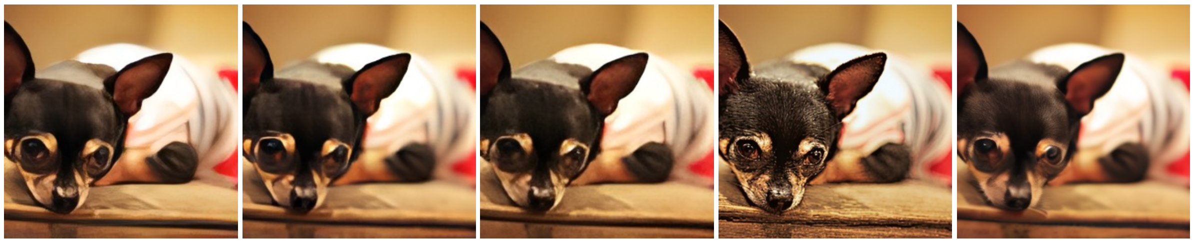
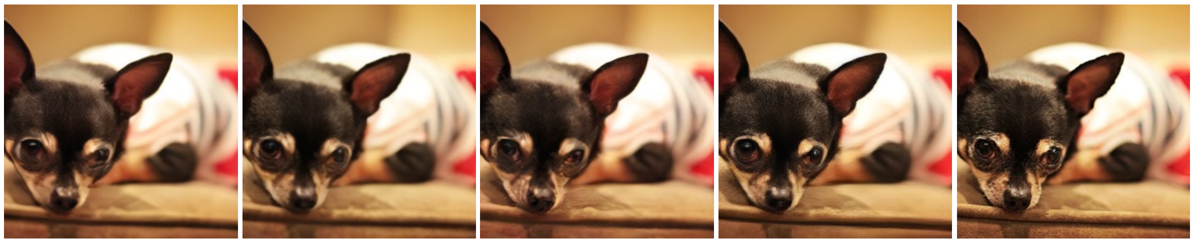
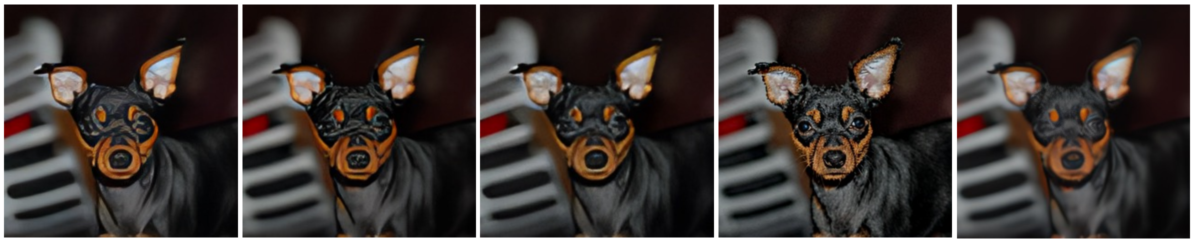
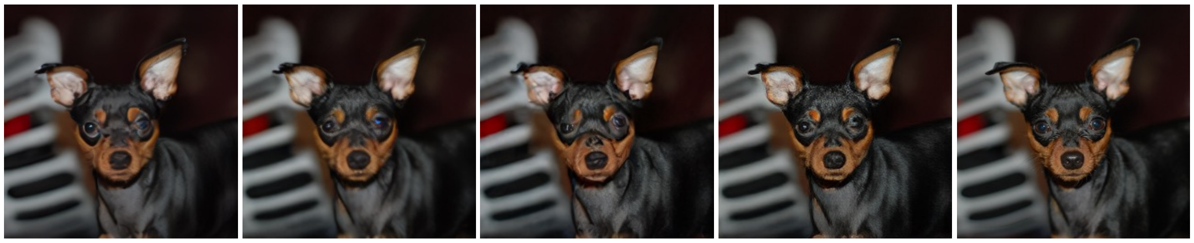


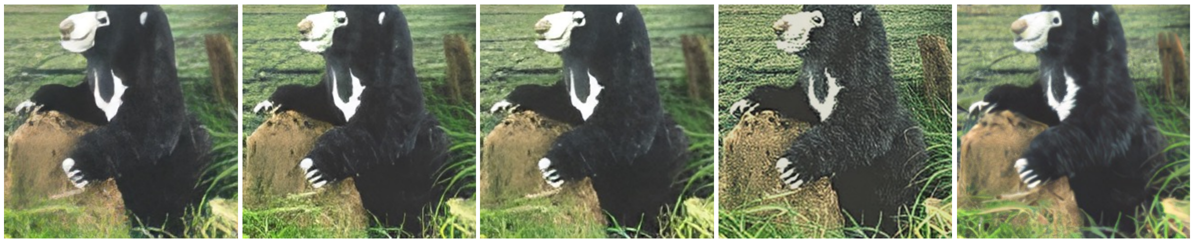
Figure 7. Visual comparison of different methods on *ImageNet-Test*. Please zoom in for more details.


(a) BSRGAN (b) SwinIR (c) RealESRGAN (d) StableSR-200 (e) LDM-15


(f) ResShift-15 (g) ResShift-4 (h) SinSR-1 (i) CTMSR-1 (j) ground-truth


(a) BSRGAN (b) SwinIR (c) RealESRGAN (d) StableSR-200 (e) LDM-15


(f) ResShift-15 (g) ResShift-4 (h) SinSR-1 (i) CTMSR-1 (j) ground-truth


(a) BSRGAN (b) SwinIR (c) RealESRGAN (d) StableSR-200 (e) LDM-15

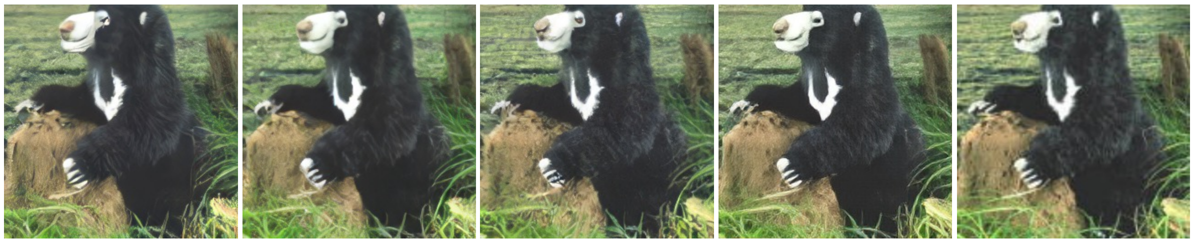
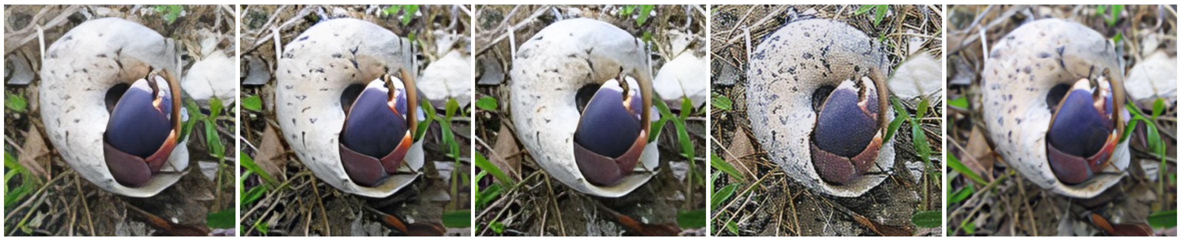
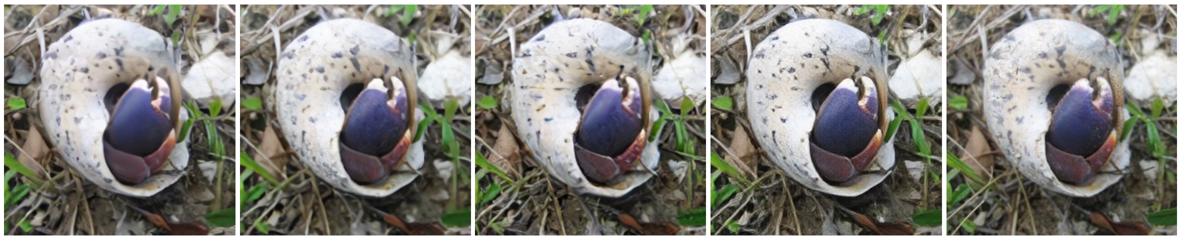

(f) ResShift-15 (g) ResShift-4 (h) SinSR-1 (i) CTMSR-1 (j) ground-truth

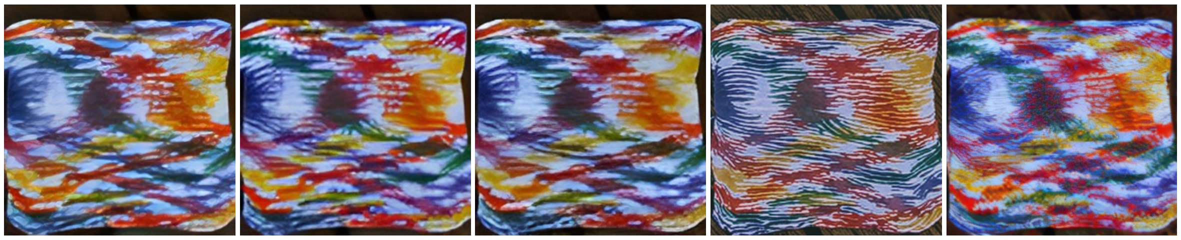
Figure 8. Visual comparison of different methods on *ImageNet-Test*. Please zoom in for more details.



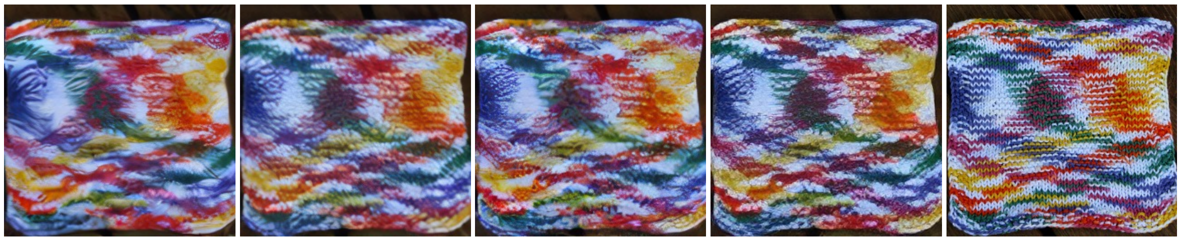
(a) BSRGAN (b) SwinIR (c) RealESRGAN (d) StableSR-200 (e) LDM-15



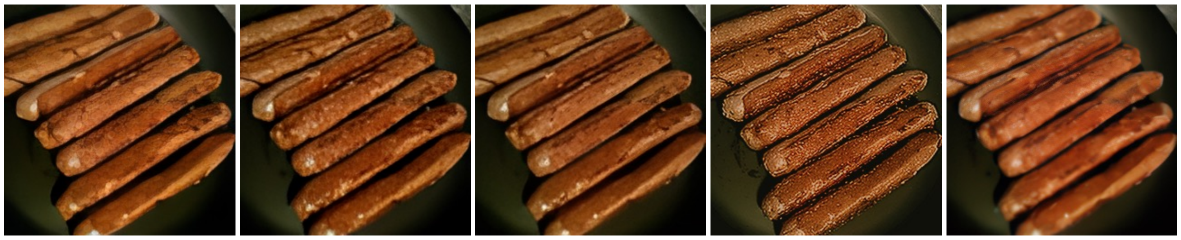
(f) ResShift-15 (g) ResShift-4 (h) SinSR-1 (i) CTMSR-1 (j) ground-truth



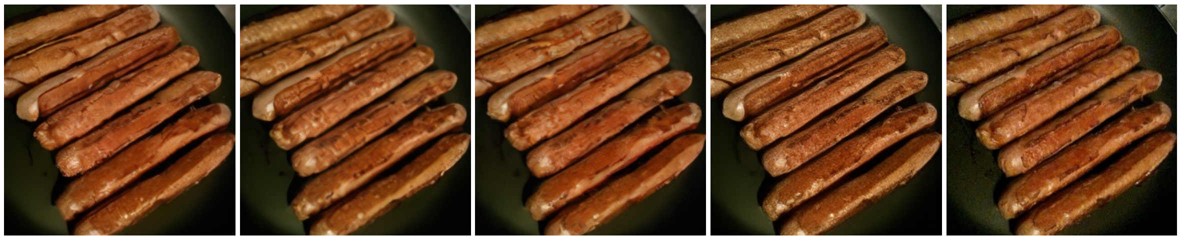
(a) BSRGAN (b) SwinIR (c) RealESRGAN (d) StableSR-200 (e) LDM-15



(f) ResShift-15 (g) ResShift-4 (h) SinSR-1 (i) CTMSR-1 (j) ground-truth



(a) BSRGAN (b) SwinIR (c) RealESRGAN (d) StableSR-200 (e) LDM-15



(f) ResShift-15 (g) ResShift-4 (h) SinSR-1 (i) CTMSR-1 (j) ground-truth

Figure 9. Visual comparison of different methods on *ImageNet-Test*. Please zoom in for more details.

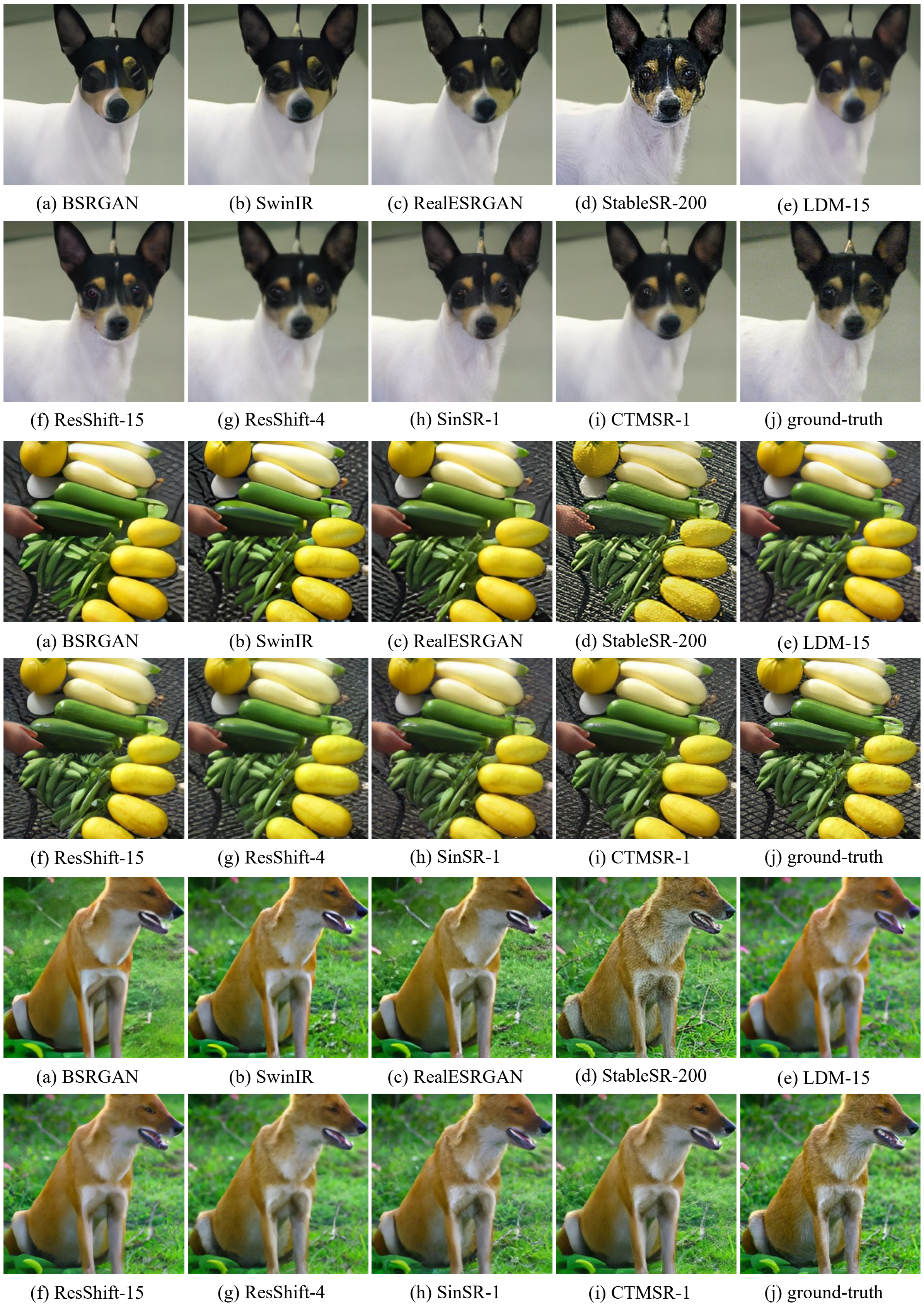


Figure 10. Visual comparison of different methods on *ImageNet-Test*. Please zoom in for more details.

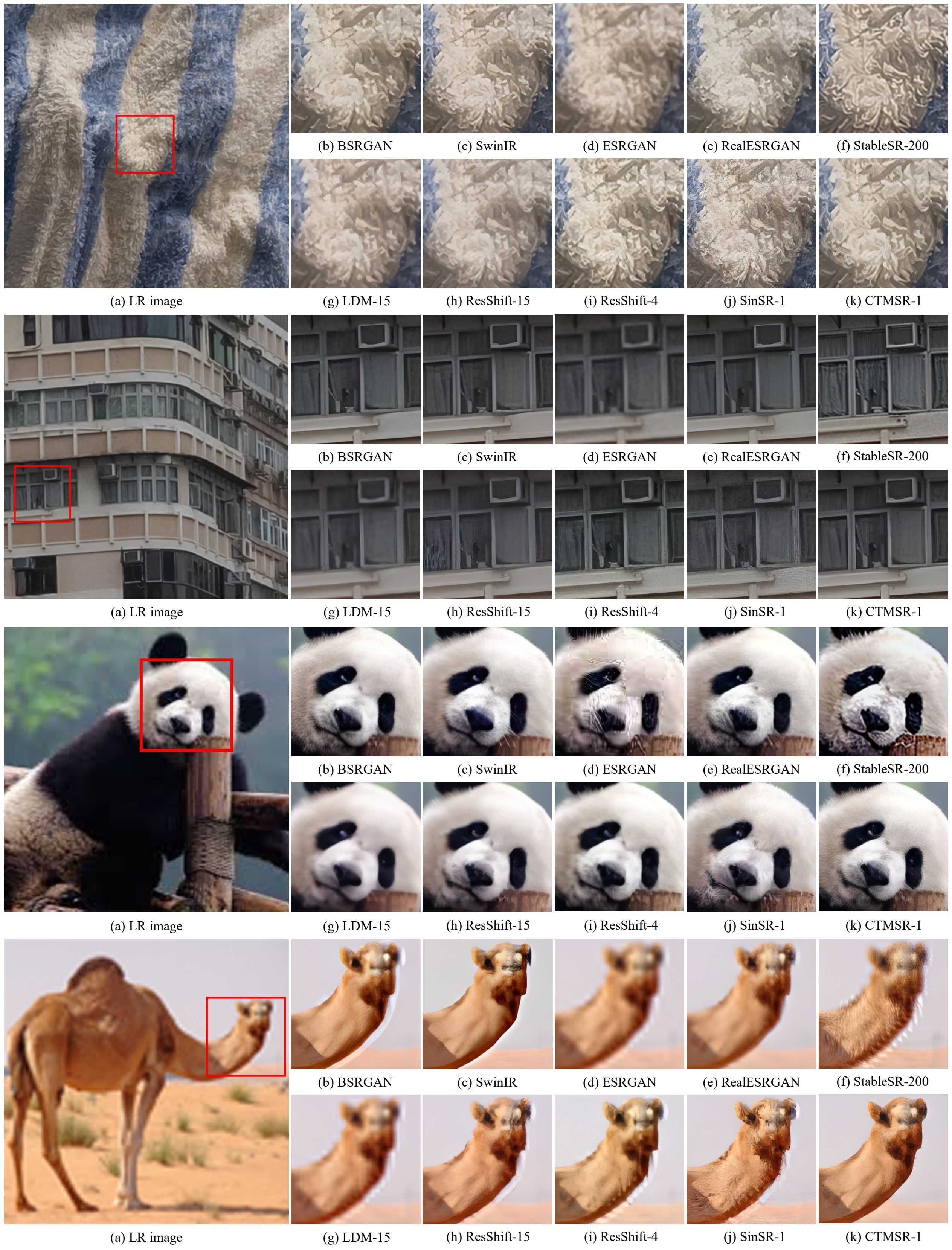
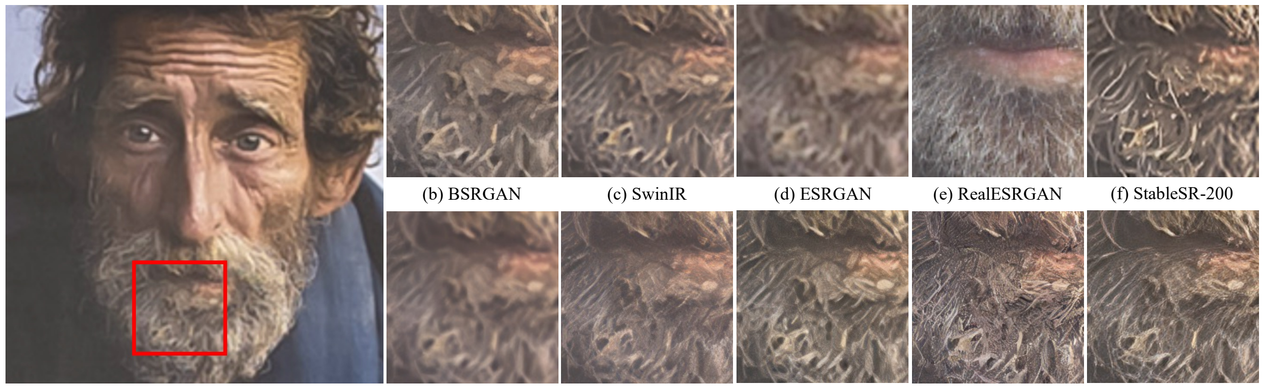
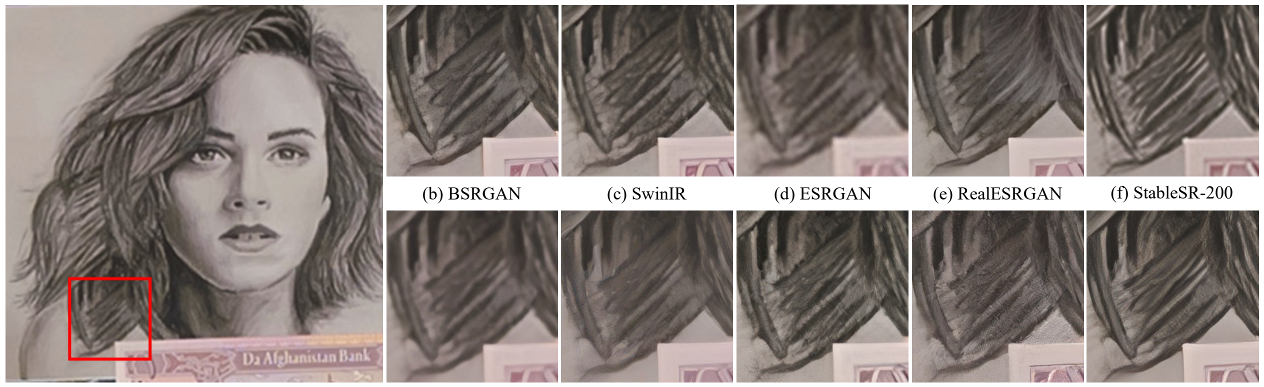
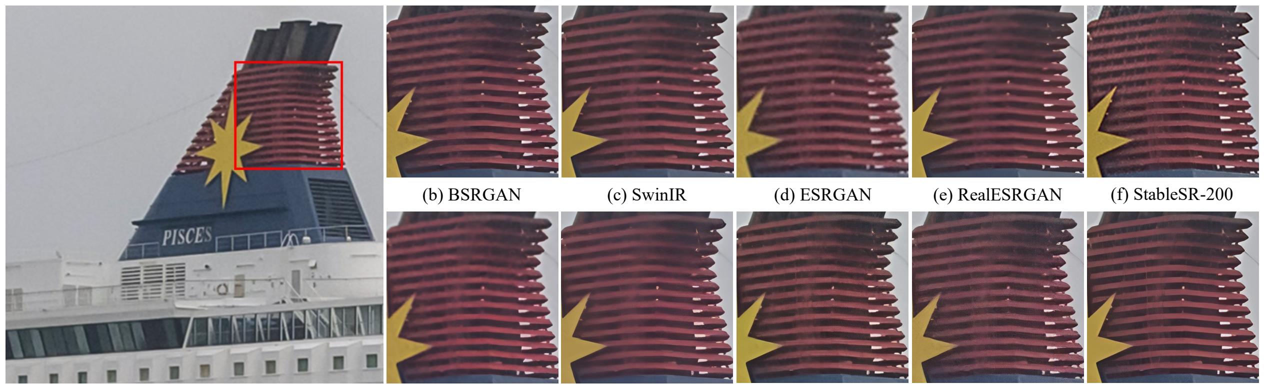


Figure 11. Visual comparison of different methods on real-world datasets. Please zoom in for more details.







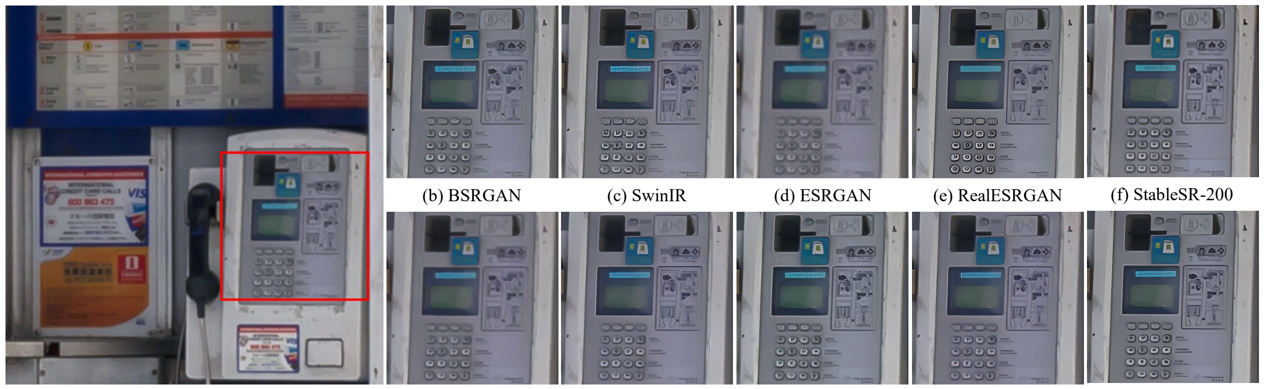


Figure 12. Visual comparison of different methods on real-world datasets. Please zoom in for more details.

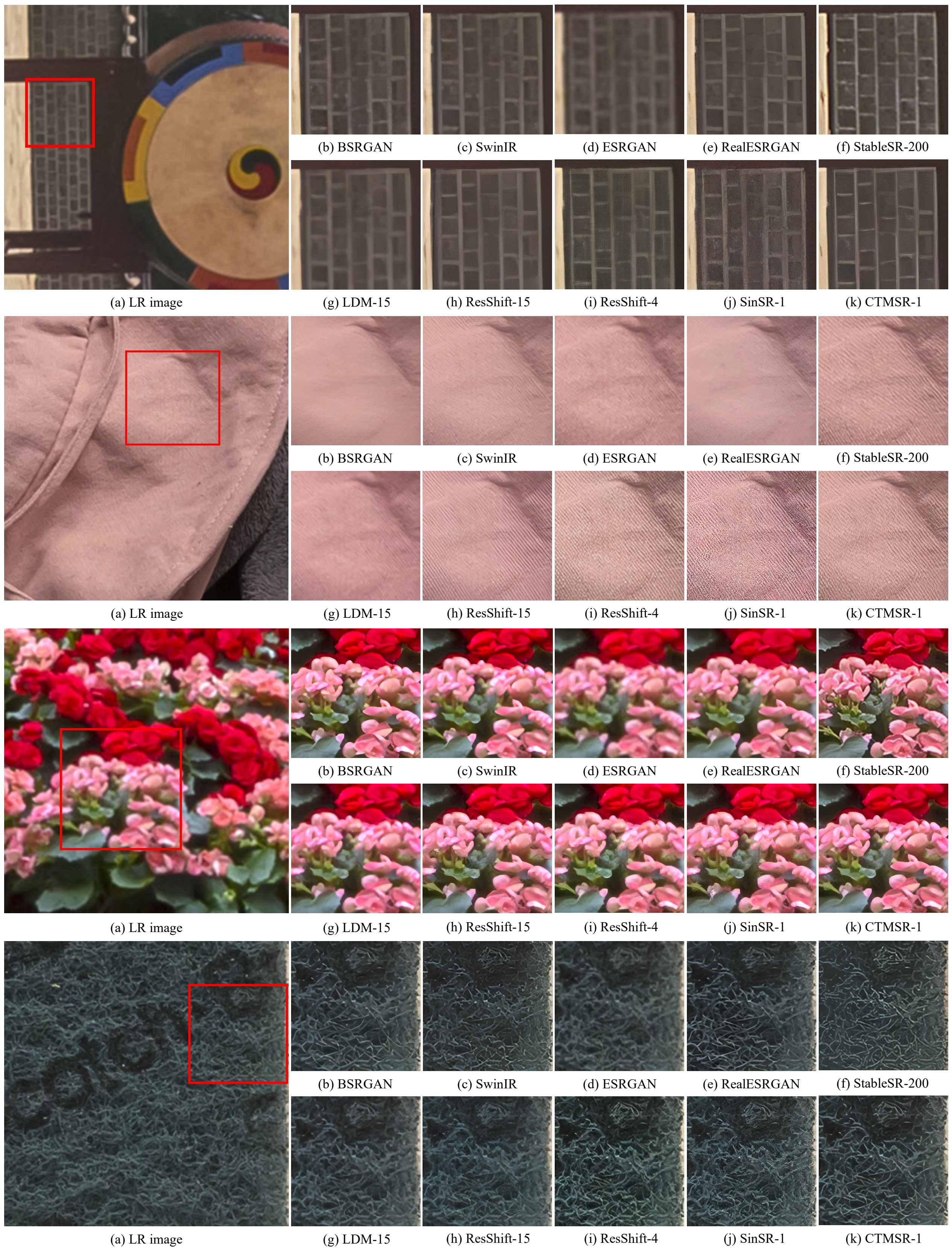


Figure 13. Visual comparison of different methods on real-world datasets. Please zoom in for more details.