Supplementary Materials: Compression-Aware One-Step Diffusion Model for JPEG Artifact Removal

Jinpei Guo[1,2], Zheng Chen[2], Wenbo Li[3], Yong Guo[4], Yulun Zhang[2*]
[1]Carnegie Mellon University, [2]Shanghai Jiao Tong University,
[3]The Chinese University of Hong Kong, [4]Max Planck Institute for Informatics

1. JPEG Compression Algorithm

JPEG [11] is one of the most widely used image compression algorithms due to its simplicity and fast encoding/decoding speeds. The JPEG algorithm applies the discrete cosine transform (DCT) to convert the image into the frequency domain. These frequency domain representations are then divided by a quantization table and rounded to the nearest integer. The quantization table elements control the compression ratio, and the rounding operation is the only lossy step in the entire process. The quantization table is typically represented by an integer known as the quality factor (QF), which ranges from 0 to 100. A lower QF results in a smaller storage size but greater information loss.

2. Double JPEG Compression

Double JPEG artifact removal aims to reconstruct the images that have been sequentially JPEG compressed twice. Specifically, double JPEG compression can be categorized into two types: aligned and non-aligned compression [5]. Aligned compression preserves the image size across two compression stages, while non-aligned compression introduces size variations between the two stages. To get non-aligned double JPEG images, we remove the first eight rows and columns of the images after the initial compression.

To demonstrate the generalization of our method, we test CODiff trained with single JPEG compression on this task. The compared methods include CNN and transformer-based methods FBCNN [5], and PromptCIR [6], with one-step diffusion methods OSEDiff [12]. All these compared methods are not exposed to double JPEG compression samples during training.

As shown in Tab. 1, our CODiff outperforms competing methods for both aligned and non-aligned double JPEG compression. Specifically, it achieves significant improvements in both full-reference metrics and no-reference metrics. These results demonstrate its superior generalization capacity to double JPEG artifact removal.

3. Extreme JPEG Compression

As stated in the main paper, our CODiff mainly focuses on highly compressed images (*e.g.*, QF=5, 10, and 20). To further validate its effectiveness, we conduct additional experiments on extreme compression reconstruction tasks, specifically at QF = 1. In Tab. 2, we present quantitative results on LIVE-1 [10], Urban100 [4], and DIV2K-Val [1] datasets. Moreover, in Fig. 2, we present visual comparisons with other competing methods.

As shown in Tab. 2, our CODiff significantly surpasses competing methods across a diverse set of evaluation metrics. The visual results in Fig.2 further highlight its remarkable performance, demonstrating its strong generalization ability in handling the challenging task of JPEG artifact removal. Furthermore, diffusion-based methods exhibit clear advantages against traditional JPEG artifact removal methods, attributed to the powerful pre-trained knowledge of T2I diffusion models [8, 9].

4. Compression-Aware Visual Embedder

Our compression-aware visual embedder (CaVE) leverages a dual learning strategy to enhance its generalization. Specifically, explicit learning is driven by the QF prediction objective, ensuring that the model learns to accurately estimate quality factors, while implicit learning focuses on the image reconstruction objective, reinforcing the model's capacity to recover visual details from compressed images.

In the main paper, we evaluate CaVE's QF prediction performance using mean squared error, particularly testing on quality factors (QFs) that are absent from the training set. In this section, we further assess its reconstruction capability on unseen QFs on LIVE-1 and DIV2K-val datasets, specifically at QF=1, 5, to demonstrate its generalization.

As presented in Tab. 3, the model trained with the dual learning strategy consistently outperforms the one relying solely on implicit learning across. This finding highlights the effectiveness of our dual learning strategy in deepening CaVE's knowledge of JPEG compression, thereby improving its ability to generalize to previously unseen QFs.

[*] Corresponding author: Yulun Zhang, yulun100@gmail.com

Methods	QF=(5, 95)					QF=(10, 90)					QF=(90, 10)				
	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑
JPEG [11]	0.4372	0.3250	40.63	0.2152	0.1282	0.2991	0.2386	54.07	0.3472	0.2042	0.3028	0.2385	53.72	0.3511	0.2737
FBCNN [5]	0.3741	0.2359	63.47	0.3410	0.2782	0.2517	0.1797	70.97	0.4182	0.4747	0.2508	0.1793	70.87	0.4186	0.4763
PromptCIR [6]	0.3842	0.2373	59.79	0.2730	0.2663	0.2320	0.1667	72.16	0.4379	0.5192	0.2301	0.1668	72.31	0.4465	0.5203
OSEDiff* [12]	0.2670	0.1653	65.47	0.3408	0.5591	0.1753	0.1175	71.25	0.3942	0.7016	0.1755	0.1174	71.11	0.3948	0.6995
CODiff (ours)	0.2078	0.1126	72.76	0.5202	0.7133	0.1422	0.0852	74.28	0.5293	0.7484	0.1427	0.0862	74.39	0.5395	0.7542

(a) Aligned double JPEG artifact removal.

Methods	QF=(5, 95)*					QF=(10, 90)*					QF=(90, 10)*				
	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑
JPEG [11]	0.4377	0.3253	42.72	0.2035	0.1213	0.3006	0.2408	55.67	0.2939	0.1676	0.3060	0.2411	55.97	0.3299	0.2886
FBCNN [5]	0.3759	0.2365	64.24	0.3365	0.2776	0.2543	0.1796	71.26	0.4143	0.4841	0.2529	0.1798	71.41	0.4151	0.4818
PromptCIR [6]	0.3888	0.2408	60.48	0.2701	0.2599	0.2469	0.1701	71.14	0.4034	0.5248	0.2316	0.1677	72.50	0.4406	0.5137
OSEDiff* [12]	0.2688	0.1649	65.91	0.3461	0.5762	0.1755	0.1158	71.16	0.3980	0.7103	0.1756	0.1159	71.14	0.3963	0.7040
CODiff (ours)	0.2096	0.1134	72.74	0.5158	0.7268	0.1479	0.0879	73.18	0.5006	0.7412	0.1433	0.0870	74.21	0.5362	0.7632

(b) Non-aligned double JPEG artifact removal.

Table 1. Quantitative results of double JPEG artifact removal on LIVE-1 dataset. QF=(QF$_1$, QF$_2$) denotes the images are first compressed with QF$_1$, and then compressed with QF$_2$. '*' denotes there is a pixel shift between two compressions. M-IQA stands for MANIQA, and C-IQA stands for CLIPIQA. The best and second best results are colored with red and blue. OSEDiff* are retrained for reference.

Methods	LIVE1-1					Urban100					DIV2K-val				
	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑	LPIPS↓	DISTS↓	MUSIQ↑	M-IQA↑	C-IQA↑
JPEG [11]	0.5516	0.4027	32.73	0.1955	0.2308	0.4548	0.3486	44.10	0.3166	0.3535	0.5484	0.4071	29.05	0.2315	0.3337
FBCNN [5]	0.5048	0.3310	51.96	0.2604	0.1495	0.3752	0.3024	60.29	0.3614	0.2487	0.4798	0.3245	43.76	0.2580	0.2221
JDEC [3]	0.5210	0.3115	44.74	0.1177	0.1806	0.4161	0.2954	52.80	0.2524	0.2696	0.5008	0.3193	39.73	0.1378	0.2343
PromptCIR [6]	0.5299	0.3711	42.92	0.1748	0.1702	0.4147	0.3184	52.27	0.3080	0.2849	0.5171	0.3722	35.09	0.1999	0.2624
DiffBIR* [7] (s=50)	0.4883	0.2852	49.43	0.2374	0.2466	0.3026	0.2231	65.52	0.3698	0.4748	0.3927	0.2167	50.23	0.2704	0.3752
SUPIR [13] (s=50)	0.5656	0.3537	38.91	0.1874	0.2188	0.4524	0.2730	62.09	0.4854	0.4462	0.5575	0.3222	41.14	0.3302	0.3165
OSEDiff* [12] (s=1)	0.4143	0.2420	56.53	0.2876	0.2384	0.3141	0.2401	64.29	0.4111	0.4255	0.3985	0.2505	50.21	0.2596	0.2829
CODiff (ours, s=1)	0.3155	0.1530	69.48	0.5039	0.6791	0.2152	0.1449	71.16	0.5737	0.6671	0.3000	0.1478	63.51	0.3944	0.6282

Table 2. Quantitative comparison on LIVE-1, Urban100 and DIV2K-Val datasets for extreme JPEG compression (*i.e.*, QF=1) reconstruction. M-IQA stands for MANIQA, and C-IQA stands for CLIPIQA. The best and second best results are colored with red and blue. DiffBIR* and OSEDiff* are retrained for reference.

Type	QF=1			QF=5		
	LPIPS↓	DISTS↓	M-IQA↑	LPIPS↓	DISTS↓	M-IQA↑
JPEG [11]	0.5516	0.4027	0.1955	0.4384	0.3242	0.2294
Implicit	0.5135	0.3100	0.1763	0.3941	0.2408	0.2393
Dual	0.4979	0.3060	0.2177	0.3795	0.2448	0.2965

(a) LIVE-1 dataset

Type	QF=1			QF-5		
	LPIPS↓	DISTS↓	M-IQA↑	LPIPS↓	DISTS↓	M-IQA↑
JPEG [11]	0.5484	0.4071	0.2315	0.4466	0.3183	0.2570
Implicit	0.4654	0.2853	0.1984	0.3516	0.2118	0.2544
Dual	0.4526	0.2826	0.2228	0.3379	0.2144	0.2877

(b) DIV2K-val Dataset

Table 3. Quantitative reconstruction results of CaVE trained via implicit and dual learning. M-IQA stands for MANIQA. The best and second best results are colored with red and blue.

5. User Study

We conduct a user study shown in Tab. 4, where 20 participants rate restored images on a 1–10 scale (higher indicates better perceptual quality). CODiff consistently achieves the highest scores across all QF levels, especially under high compression, demonstrating its superior perceptual quality and robustness compared to prior methods.

Method	QF=1	QF=5	QF=10	QF=20	QF=80	QF=90	QF=100
DiffBIR [7]	2.88	5.32	6.62	7.87	9.22	9.41	9.85
OSEDiff [12]	4.07	5.54	6.73	7.98	9.15	9.63	9.79
CODiff (ours)	5.22	6.37	7.79	8.49	9.44	9.87	9.94

Table 4. Average user ratings across methods. The best results are colored with red.

6. CaVE for multi-step diffusion

We incorporate CaVE into the 50-step DiffBIR model by concatenating its compression prior with the LQ condition and retraining the model. As shown in Tab. 5, CaVE brings consistent improvements across all metrics and QFs, demonstrating its effectiveness in enhancing multi-step diffusion models with compression-aware priors.

Method	QF=5			QF=10			QF=20		
	LPIPS↓	MUSIQ↑	M-IQA↑	LPIPS↓	MUSIQ↑	M-IQA↑	LPIPS↓	MUSIQ↑	M-IQA↑
w/o CaVE	0.2788	60.21	0.3220	0.1953	65.22	0.3754	0.1542	67.06	0.4033
w/ CaVE	0.2716	65.37	0.3713	0.1895	69.65	0.4209	0.1488	71.59	0.4478

Table 5. Effect of CaVE on DiffBIR (DIV2K-val dataset). The best results are colored with red.

7. Performance on other compression formats

We conduct experiments on BPG [2] by retraining CODiff and two leading baselines under this setting. As shown in Tab. 6, CODiff consistently outperforms prior methods across all metrics and QP levels. Notably, it achieves substantial gains under challenging high-QP settings (e.g., QP=50), demonstrating strong generalization and effectiveness in removing BPG artifacts.

Method	QP=50			QP=40			QP=30		
	DISTS↓	LPIPS↓	C-IQA↑	DISTS↓	LPIPS↓	C-IQA↑	DISTS↓	LPIPS↓	C-IQA↑
DiffBIR [7]	0.1684	0.2806	0.4899	0.0958	0.1266	0.5724	0.0407	0.0472	0.6230
OSEDiff [12]	0.1571	0.2663	0.4752	0.0907	0.1184	0.5910	0.0373	0.0356	0.6441
CODiff (ours)	0.1083	0.1932	0.6454	0.0717	0.0644	0.6881	0.0266	0.0273	0.7183

Table 6. BPG artifact removal results on DIV2K-val dataset. The best results are colored with red.

8. Additional Visual Comparisons

In this part, we provide more visual comparisons between CODiff and other baseline methods. The compared methods are: (1) CNN and transformer-based methods, including FBCNN [5], JDEC [3], and PromptCIR [6]. (2) Diffusion-based methods, including DIffBIR [7], SUPIR [13], and OSEDiff [12]. To highlight the restoration capabilities of CODiff, we present visual results across different compression levels. Compared to existing methods, CODiff demonstrates outstanding performance in recovering intricate details and generating photorealistic images across various compression settings. Notably, as shown in Fig. 2, even under the extreme compression condition (*i.e.*, QF=1), CODiff still effectively reconstructs architectural details, such as windows, grid lines, and bricks. These results showcase CODiff's superiority for highly compressed images.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017. 1

[2] Fabrice Bellard. Bpg image format. https://bellard.org/bpg/, 2014. Accessed: 2025-05-08. 2

[3] Woo Kyoung Han, Sunghoon Im, Jaedeok Kim, and Kyong Hwan Jin. Jdec: Jpeg decoding via enhanced continuous cosine coefficients. In *CVPR*, 2024. 2, 3, 4, 5, 6, 7, 8, 9

[4] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 1

[5] Jiaxi Jiang, Kai Zhang, and Radu Timofte. Towards flexible blind jpeg artifacts removal. In *ICCV*, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9

[6] Bingchen Li, Xin Li, Yiting Lu, Ruoyu Feng, Mengxi Guo, Shijie Zhao, Li Zhang, and Zhibo Chen. Promptcir: Blind compressed image restoration with prompt learning. *arXiv preprint arXiv:2404.17433*, 2024. 1, 2, 3, 4, 5, 6, 7, 8, 9

[7] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *ECCV*, 2024. 2, 3, 4, 5, 6, 7, 8, 9

[8] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[10] H Sheikh. Live image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*, 2005. 1

[11] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 1991. 1, 2

[12] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *arXiv preprint arXiv:2406.08177*, 2024. 1, 2, 3, 4, 5, 6, 7, 8, 9

[13] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. In *CVPR*, 2024. 2, 3

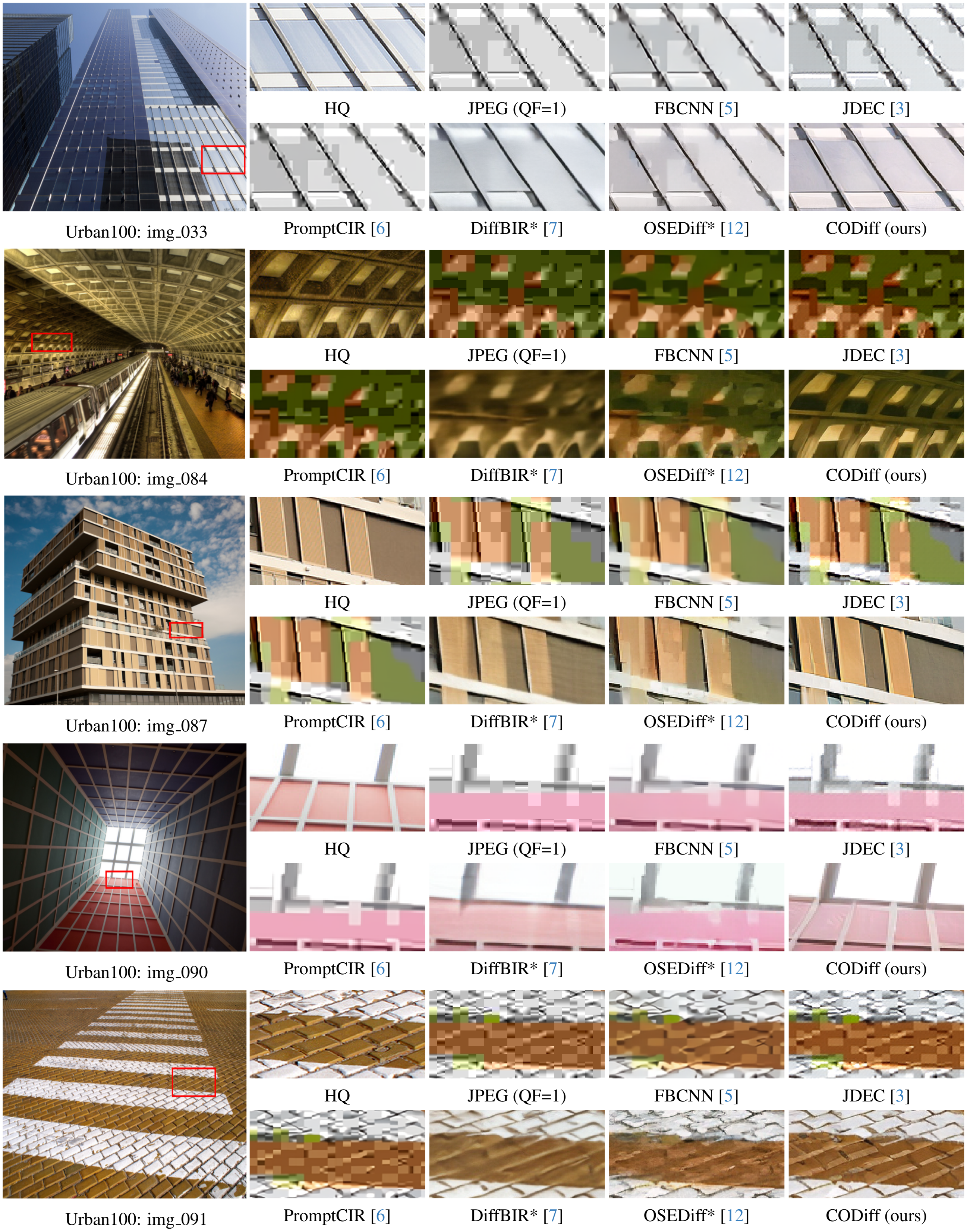


Figure 1. More visual comparisons (QF=1) between CODiff and other competing methods. Please zoom in for a better view.

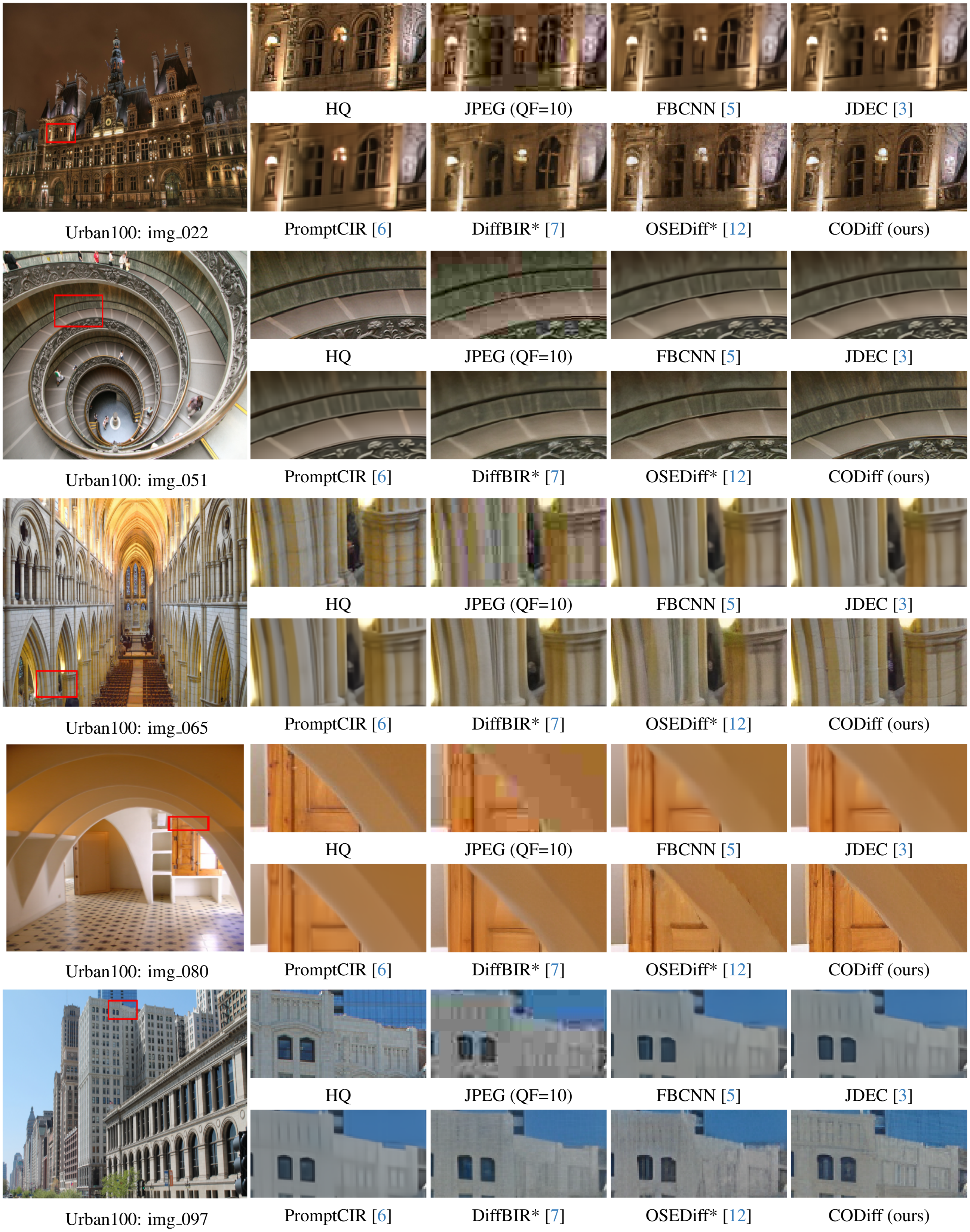
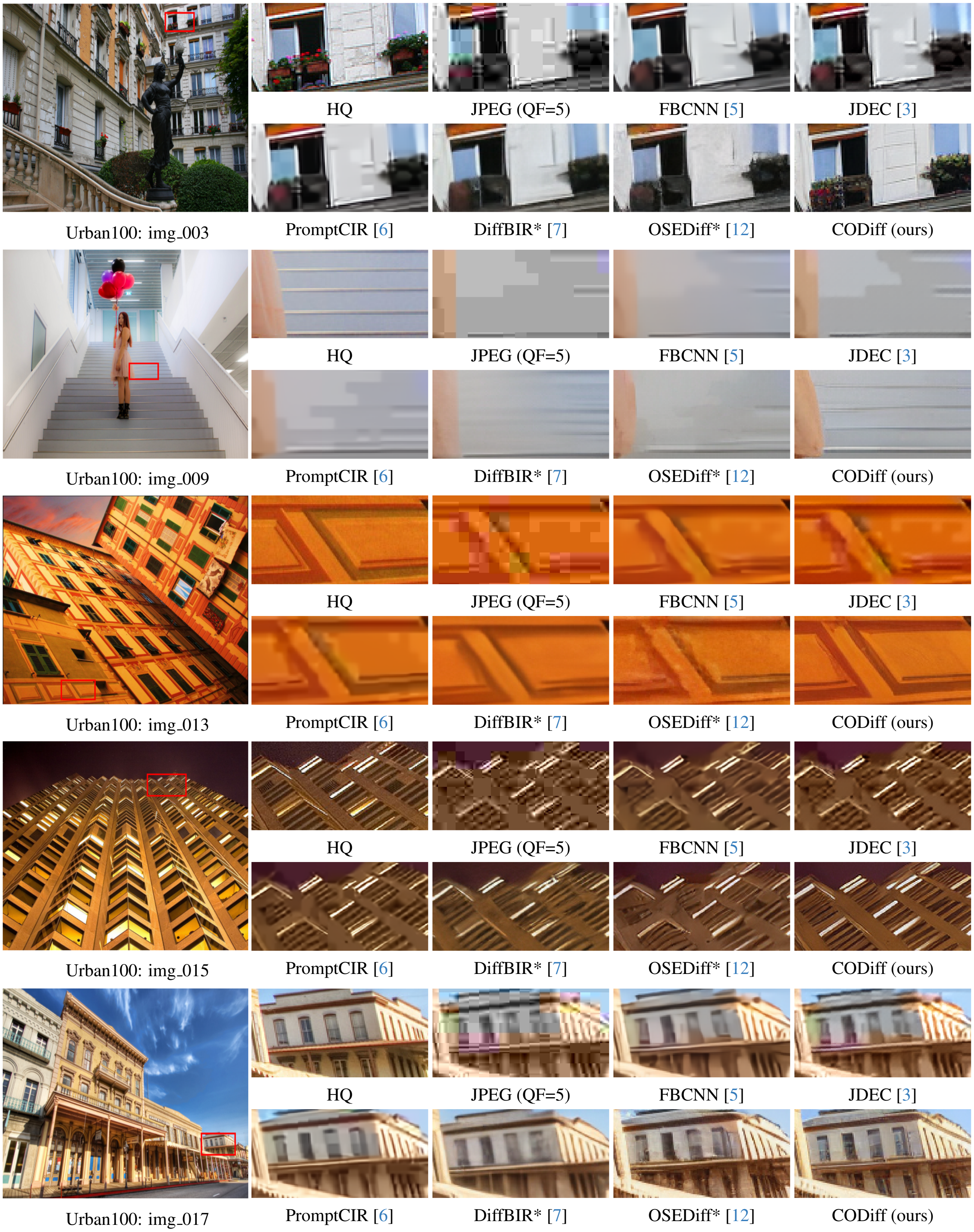


Figure 2. More visual comparisons (QF=10) between CODiff and other competing methods. Please zoom in for a better view.



Urban100: img_003

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_009

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_013

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_015

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_017

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Figure 3. More visual comparisons (QF=5) between CODiff and other competing methods. Please zoom in for a better view.

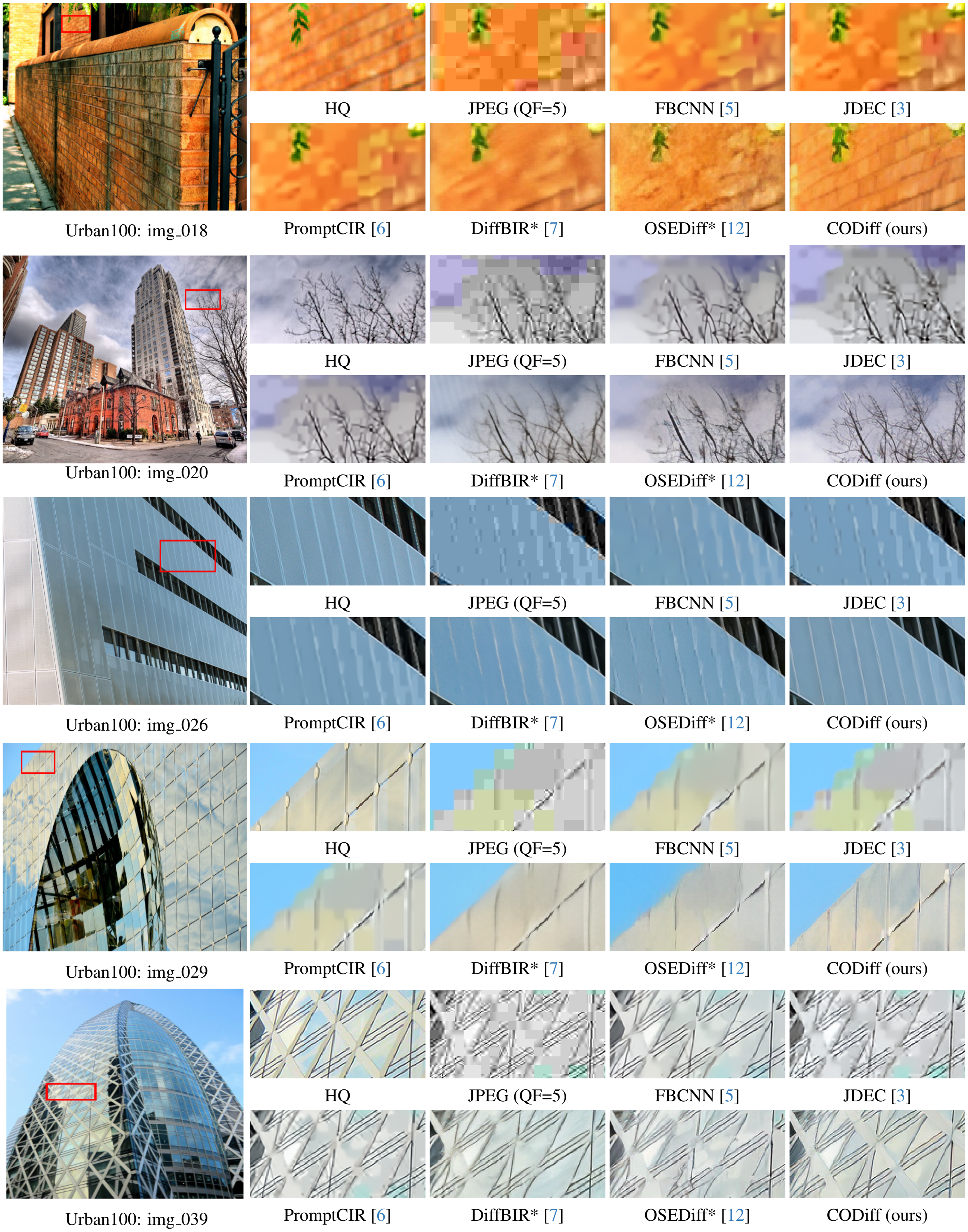


Figure 4. More visual comparisons (QF=5) between CODiff and other competing methods. Please zoom in for a better view.

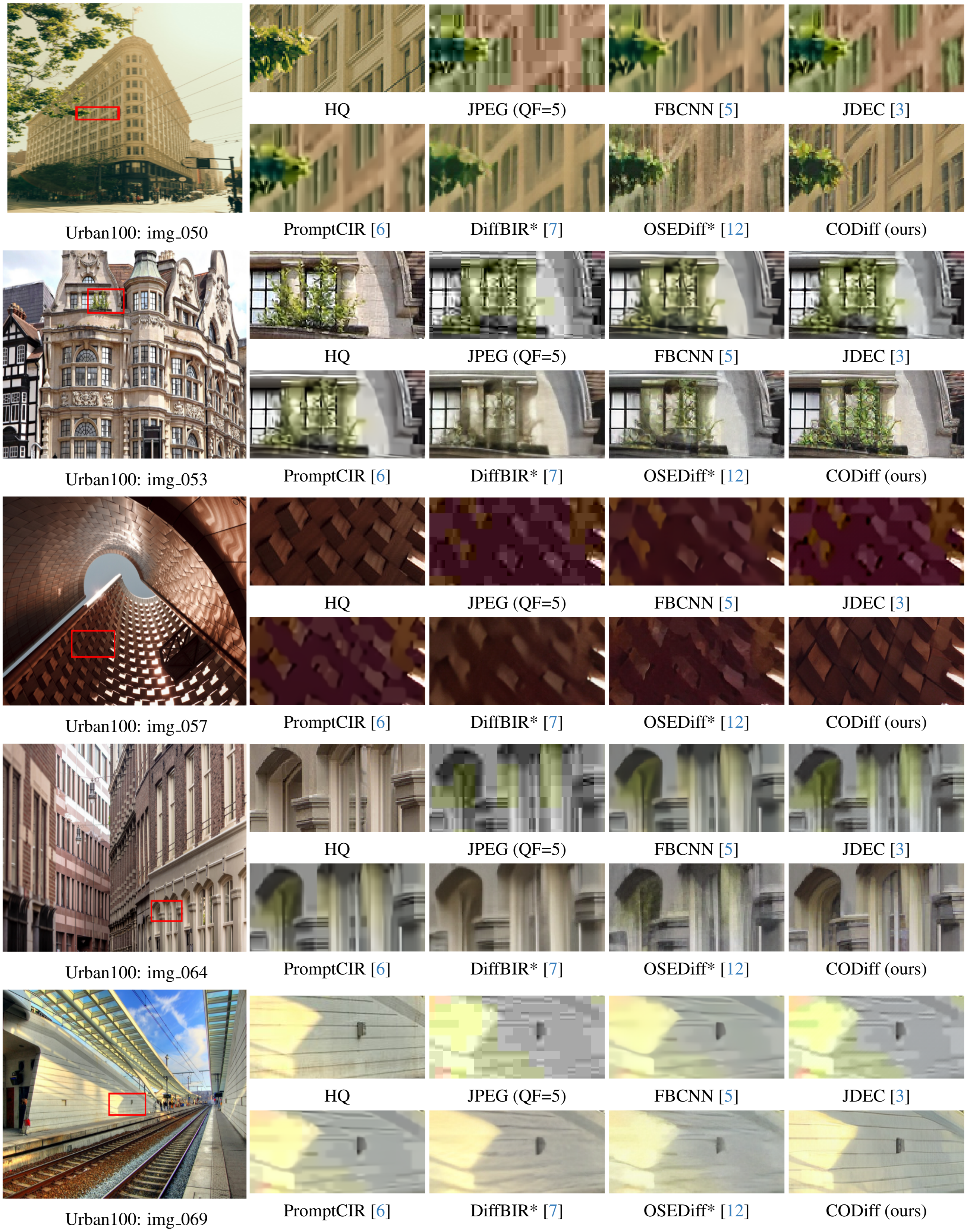
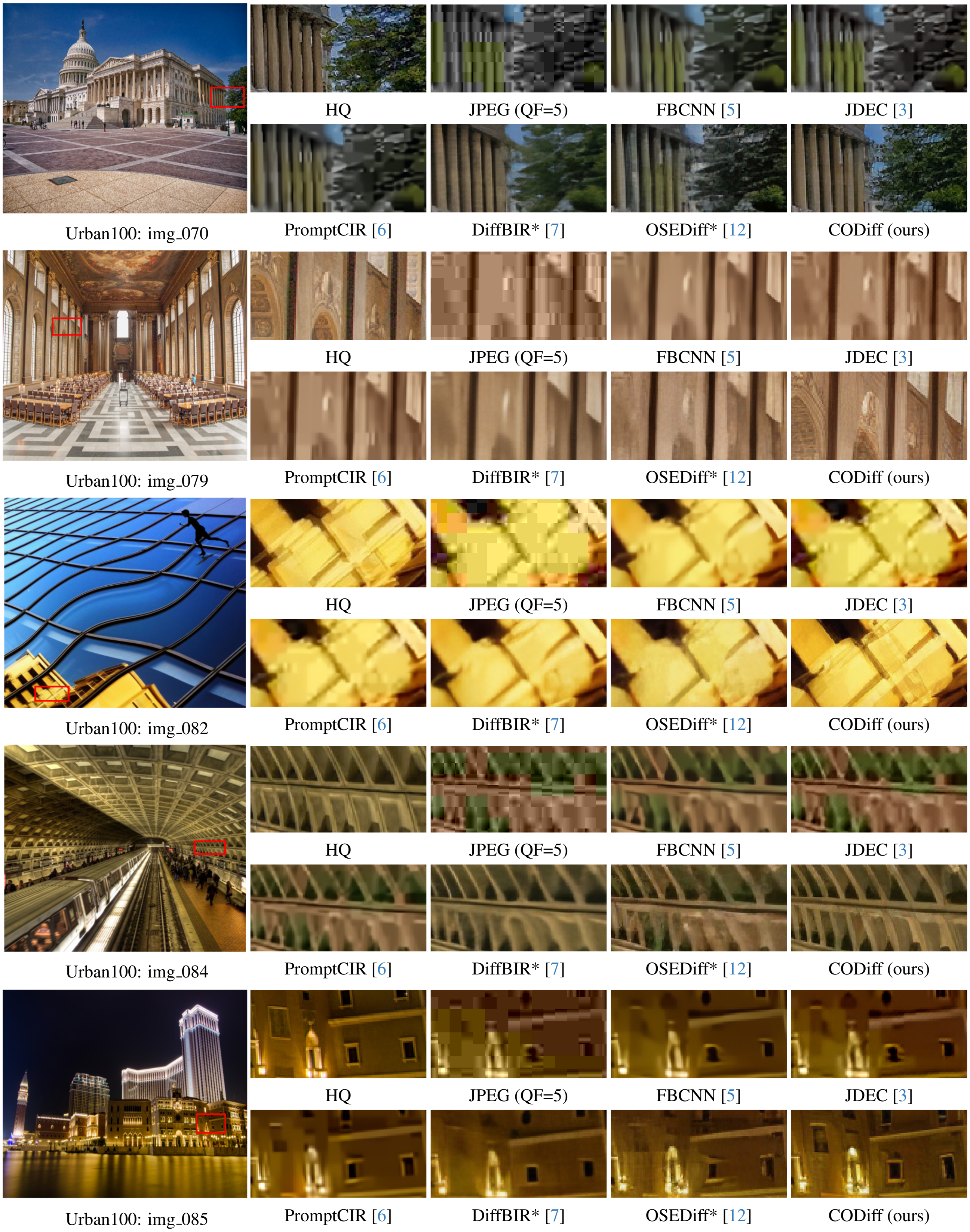


Figure 5. More visual comparisons (QF=5) between CODiff and other competing methods. Please zoom in for a better view.



Urban100: img_070

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_079

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_082

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_084

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Urban100: img_085

HQ | JPEG (QF=5) | FBCNN [5] | JDEC [3]

PromptCIR [6] | DiffBIR* [7] | OSEDiff* [12] | CODiff (ours)

Figure 6. More visual comparisons (QF=5) between CODiff and other competing methods. Please zoom in for a better view.