

Supplementary File: Image Super-Resolution Using Very Deep Residual Channel Attention Networks

Yulun Zhang¹, Kunpeng Li¹, Kai Li¹, Lichen Wang¹,
Bineng Zhong¹, and Yun Fu^{1,2}

¹Department of ECE, Northeastern University, Boston, USA

²College of Computer and Information Science, Northeastern University, Boston, USA

Abstract. In this supplementary file, we first provide convergence analyses of three very deep networks to validate the effectiveness of our proposed residual in residual structure. Then we compare with over 20 state-of-the-art super-resolution methods with bicubic (BI) and blur-down (BD) degradation models. For BI model, we not only compare with generative models for $4\times$ and $8\times$ SR, but also show comparisons with generative adversarial networks based SR methods. For BD model, we provide more quantitative and visual results. For both BI and DB models, we compare state-of-the-art methods with/without self-ensemble to further show the improvements of our proposed method. All quantitative results are evaluated in terms of PSNR and SSIM on standard datasets.

1 Experimental Results

1.1 Convergence Analyses

We first conduct analyses on three very deep residual networks, which have 400 residual blocks (RB). As shown in Figure 1, the green line (R400_SS) denotes the network simply stacking 400 RBs. The blue line (R400_LSC) denotes the network with long skip connection (LSC). This can be viewed as MDSR [1], whose RB number is set as 400 and scaling factor is set as 2. The red line (R400_RIR) denotes the network that uses our proposed residual in residual (RIR) structure.

All these three networks are trained from scratch. To save the training time, here we set batch size as 16 and input size as 32×32 (different from 48×48 in our main manuscript). From Figure 1, we mainly have three observations and corresponding explanations:

(1) Simply stacking lots of RBs to construct very deep networks is not applicable for image SR. We can see the green line (R400_SS) would start at a relatively low performance. Its training process is not stable. The performance after training 200 epochs is relatively low either. The main reason causing the result is that information flow could become more difficult in deeper networks. Furthermore, spatial information is very important for image SR. While, the tail convolutional (Conv) layers in very deep network would extract more high-level features, which is not sufficient for image SR.

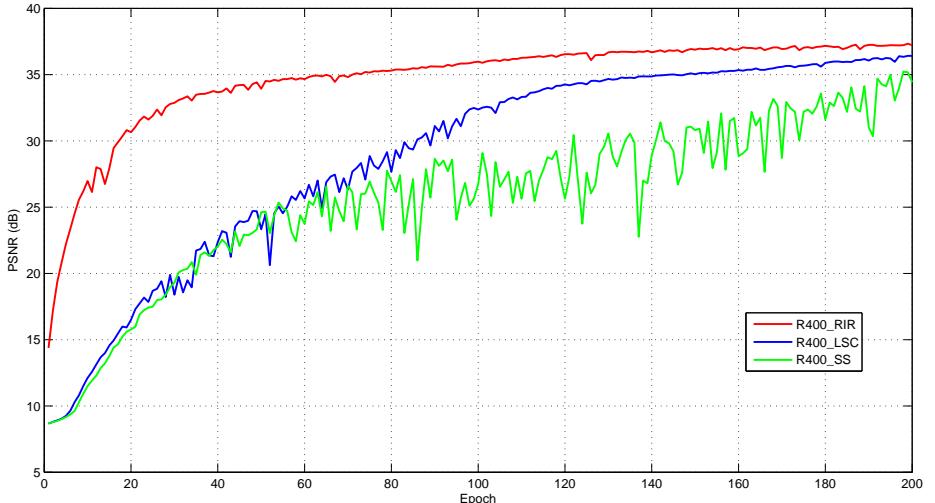


Fig. 1. Convergence analyses on three very deep residual networks. All the three networks have 400 residual blocks (RB). The green line (R400_SS) denotes the network simply stacking 400 RBs. The blue line (R400_LSC) denotes the network with long skip connection (LSC). This can be viewed as MDSR [1], whose RB number is set as 400. The red line (R400_RIR) denotes the network that uses our proposed RIR structure. The curves are based on the PSNR values on Set5 ($2\times$) in 200 epochs

(2) MDSR [1] with very large number of RBs would also hardly obtain benefits from very large network depth. We can also see the blue line (R400_LSC) would start at a relatively low performance, being similar as that of R400_SS. The training process of R400_LSC is more stable than that of R400_SS. R400_LSC also has higher PSNR value than that of R400_SS after 200 epochs. This difference is mainly made by introducing LSC, which not only helps information flow, but also forwards low-level features to the tail Conv layers.

(3) Very deep networks with RIR structure has better performance and its training process is more stable. We can see the red line (R400_RIR) would start at a relatively high performance. It converges much faster than R400_SS and R400_LSC. After training 200 epochs, R400_RIR achieves the highest performance. These improvements mainly come from our proposed RIR structure, which encourages the information flow and forwards features of different levels to the tail Conv layers.

In summary, the experiments above show that simply stacking RBs or introducing LSC would not be applicable for very deep networks. To construct very deep trainable network and obtain benefits from very large network depth, our proposed RIR structure can be a good choice. The following experiments would further demonstrate the superior performance of our proposed residual channel attention network (RCAN) against other state-of-the-art methods.

Table 1. Quantitative results with BI model. Self-ensemble is **NOT** used to further enhance the results. Best and second best results are **highlighted** and underlined

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM								
Bicubic	$\times 2$	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SelfExSR [2]	$\times 2$	36.50	0.9537	32.23	0.9036	31.18	0.8855	29.38	0.9032	35.82	0.9690
SRCNN [3]	$\times 2$	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.50	0.8946	35.60	0.9663
FSRCNN [4]	$\times 2$	37.05	0.9560	32.66	0.9090	31.53	0.8920	29.88	0.9020	36.67	0.9710
SCN [5]	$\times 2$	36.58	0.9540	32.36	0.9036	31.22	0.8833	29.47	0.8953	35.34	0.9654
VDSR [6]	$\times 2$	37.53	0.9590	33.05	0.9130	31.90	0.8960	30.77	0.9140	37.22	0.9750
LapSRN [7]	$\times 2$	37.52	0.9591	33.08	0.9130	31.08	0.8950	30.41	0.9101	37.27	0.9740
DRRN [8]	$\times 2$	37.74	0.9590	33.23	0.9140	32.05	0.8970	31.23	0.9190	37.92	0.9760
MemNet [9]	$\times 2$	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
EDSR [1]	$\times 2$	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
MDSR [1]	$\times 2$	38.11	0.9602	33.85	0.9198	32.29	0.9007	32.84	0.9347	38.96	0.9769
SRMDNF [10]	$\times 2$	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
D-DBPN [11]	$\times 2$	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
RDN [12]	$\times 2$	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
RCAN (ours)	$\times 2$	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
Bicubic	$\times 3$	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	26.95	0.8556
SelfExSR [2]	$\times 3$	32.62	0.9094	29.16	0.8197	28.30	0.7843	26.45	0.8100	27.57	0.8210
SRCNN [3]	$\times 3$	32.75	0.9090	29.30	0.8215	28.41	0.7863	26.24	0.7989	30.48	0.9117
FSRCNN [4]	$\times 3$	33.18	0.9140	29.37	0.8240	28.53	0.7910	26.43	0.8080	31.10	0.9210
SCN [5]	$\times 3$	32.59	0.9080	29.20	0.8175	28.31	0.7809	26.19	0.7995	30.12	0.9110
VDSR [6]	$\times 3$	33.67	0.9210	29.78	0.8320	28.83	0.7990	27.14	0.8290	32.01	0.9340
LapSRN [7]	$\times 3$	33.82	0.9227	29.87	0.8320	28.82	0.7980	27.07	0.8280	32.21	0.9350
DRRN [8]	$\times 3$	34.03	0.9240	29.96	0.8350	28.95	0.8000	27.53	0.7640	32.74	0.9390
MemNet [9]	$\times 3$	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	32.51	0.9369
EDSR [1]	$\times 3$	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	<u>34.17</u>	0.9476
MDSR [1]	$\times 3$	34.66	0.9280	30.44	0.8452	29.25	0.8091	28.79	0.8655	34.17	0.9473
SRMDNF [10]	$\times 3$	34.12	0.9254	30.04	0.8382	28.97	0.8025	27.57	0.8398	33.00	0.9403
RDN [12]	$\times 3$	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484
RCAN (ours)	$\times 3$	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
Bicubic	$\times 4$	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SelfExSR [2]	$\times 4$	30.33	0.8623	27.40	0.7518	26.85	0.7108	24.82	0.7386	27.83	0.8660
SRCNN [3]	$\times 4$	30.48	0.8628	27.50	0.7513	26.90	0.7101	24.52	0.7221	27.58	0.8555
FSRCNN [4]	$\times 4$	30.72	0.8660	27.61	0.7550	26.98	0.7150	24.62	0.7280	27.90	0.8610
SCN [5]	$\times 4$	30.39	0.8628	27.45	0.7501	26.86	0.7082	24.51	0.7239	27.32	0.8535
VDSR [6]	$\times 4$	31.35	0.8830	28.02	0.7680	27.29	0.7026	25.18	0.7540	28.83	0.8870
LapSRN [7]	$\times 4$	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560	29.09	0.8900
DRRN [8]	$\times 4$	31.68	0.8880	28.21	0.7720	27.38	0.7280	25.44	0.7640	29.46	0.8960
SRResNet [13]	$\times 4$	32.05	0.9019	28.49	0.8184	27.58	0.7620	26.07	0.7839	N/A	N/A
SRGAN [13]	$\times 4$	29.40	0.8472	26.02	0.7397	25.16	0.6688	N/A	N/A	N/A	N/A
MemNet [9]	$\times 4$	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
SRDenseNet [14]	$\times 4$	32.02	0.8930	28.50	0.7780	27.53	0.7337	26.05	0.7819	N/A	N/A
ENet-E [15]	$\times 4$	31.74	0.8869	28.42	0.7774	27.50	0.7326	25.66	0.7703	N/A	N/A
ENet-PAT [15]	$\times 4$	28.56	0.8082	25.77	0.6784	24.93	0.6270	23.54	0.6936	N/A	N/A
EDSR [1]	$\times 4$	32.46	0.8968	28.80	<u>0.7876</u>	27.71	0.7420	26.64	0.8033	31.02	0.9148
MDSR [1]	$\times 4$	32.50	0.8973	28.72	0.7857	27.72	0.7418	<u>26.67</u>	<u>0.8041</u>	31.11	0.9148
SRMDNF [10]	$\times 4$	31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	<u>0.7731</u>	30.09	0.9024
D-DBPN [11]	$\times 4$	32.47	0.8980	<u>28.82</u>	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
RDN [12]	$\times 4$	32.47	0.8990	28.81	0.7871	<u>27.72</u>	<u>0.7419</u>	26.61	0.8028	31.00	<u>0.9151</u>
RCAN (ours)	$\times 4$	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
Bicubic	$\times 8$	24.40	0.6580	23.10	0.5660	23.67	0.5480	20.74	0.5160	21.47	0.6500
SelfExSR [2]	$\times 8$	25.49	0.7030	23.92	0.6101	24.19	0.5680	21.81	0.5770	22.99	0.7190
SRCNN [3]	$\times 8$	25.33	0.6900	23.76	0.5910	24.13	0.5660	21.29	0.5440	22.46	0.6950
FSRCNN [4]	$\times 8$	20.13	0.5520	19.75	0.4820	24.21	0.5680	21.32	0.5380	22.39	0.6730
SCN [5]	$\times 8$	25.59	0.7071	24.02	0.6028	24.30	0.5698	21.52	0.5571	22.68	0.6963
VDSR [6]	$\times 8$	25.93	0.7240	24.26	0.6140	24.49	0.5830	21.70	0.5710	23.16	0.7250
LapSRN [7]	$\times 8$	26.15	0.7380	24.35	0.6200	24.54	0.5860	21.81	0.5810	23.39	0.7350
MemNet [9]	$\times 8$	26.16	0.7414	24.38	0.6199	24.58	0.5842	21.89	0.5825	23.56	0.7387
MSLapSRN [16]	$\times 8$	26.34	0.7558	24.57	0.6273	24.65	0.5895	22.06	0.5963	23.90	0.7564
EDSR [1]	$\times 8$	26.96	0.7762	24.91	0.6420	24.81	0.5985	22.51	0.6221	24.69	0.7841
MDSR [1]	$\times 8$	26.97	0.7764	24.90	0.6416	24.76	0.5972	22.39	0.6175	24.55	0.7822
D-DBPN [11]	$\times 8$	<u>27.21</u>	<u>0.7840</u>	<u>25.13</u>	<u>0.6480</u>	<u>24.88</u>	<u>0.6010</u>	<u>22.73</u>	<u>0.6312</u>	<u>25.14</u>	<u>0.7987</u>
RCAN (ours)	$\times 8$	27.31	0.7878	25.23	0.6511	24.98	0.6058	23.00	0.6452	25.24	0.8029

Table 2. Quantitative results with BI model. Self-ensemble is used to further enhance results except for RCAN. Best and second best results are **highlighted** and underlined

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM								
Bicubic	$\times 2$	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
EDSR+ [1]	$\times 2$	38.20	0.9606	34.02	0.9204	32.37	0.9018	33.10	0.9363	39.28	0.9776
MDSR+ [1]	$\times 2$	38.17	0.9605	33.92	0.9203	32.34	0.9014	33.03	0.9362	39.16	0.9774
RDN+ [12]	$\times 2$	<u>38.30</u>	<u>0.9616</u>	34.10	<u>0.9218</u>	32.40	0.9022	33.09	0.9368	39.38	0.9784
RCAN (ours)	$\times 2$	38.27	0.9614	34.12	0.9216	32.41	<u>0.9027</u>	33.34	0.9384	39.44	0.9786
RCAN+ (ours)	$\times 2$	38.33	0.9617	34.23	0.9225	32.46	0.9031	33.54	0.9399	39.61	0.9788
Bicubic	$\times 3$	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	26.95	0.8556
EDSR+ [1]	$\times 3$	34.76	0.9290	30.66	0.8481	29.32	0.8104	29.02	0.8685	34.52	0.9493
MDSR+ [1]	$\times 3$	34.77	0.9288	30.53	0.8465	29.30	0.8101	28.99	0.8683	34.43	0.9486
RDN+ [12]	$\times 3$	34.78	0.9300	30.67	0.8482	29.33	0.8105	29.00	0.8683	34.43	0.9498
RCAN (ours)	$\times 3$	34.74	0.9299	30.65	0.8482	29.32	<u>0.8111</u>	29.09	<u>0.8702</u>	34.44	0.9499
RCAN+ (ours)	$\times 3$	34.85	0.9305	30.76	0.8494	29.39	0.8122	29.31	0.8736	34.76	0.9513
Bicubic	$\times 4$	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
EDSR+ [1]	$\times 4$	32.62	0.8984	28.94	0.7901	27.79	<u>0.7437</u>	26.86	0.8080	31.43	0.9182
MDSR+ [1]	$\times 4$	32.61	0.8982	28.82	0.7876	27.78	0.7425	26.86	0.8082	31.42	0.9175
RDN+ [12]	$\times 4$	32.61	0.9003	28.92	0.7893	27.80	0.7434	26.82	0.8069	31.39	0.9184
RCAN (ours)	$\times 4$	<u>32.63</u>	0.9002	28.87	0.7889	27.77	<u>0.7436</u>	<u>26.82</u>	<u>0.8087</u>	31.22	0.9173
RCAN+ (ours)	$\times 4$	32.73	0.9013	28.98	0.7910	27.85	0.7455	27.10	0.8142	31.65	0.9208
Bicubic	$\times 8$	24.40	0.6580	23.10	0.5660	23.67	0.5480	20.74	0.5160	21.47	0.6500
EDSR+ [1]	$\times 8$	27.19	0.7835	25.12	0.6467	24.90	0.6011	22.71	0.6284	24.97	0.7915
MDSR+ [1]	$\times 8$	27.20	0.7847	25.08	0.6461	24.89	0.6007	22.71	0.6281	24.94	0.7910
RCAN (ours)	$\times 8$	<u>27.31</u>	<u>0.7878</u>	25.23	<u>0.6511</u>	24.98	<u>0.6058</u>	<u>23.00</u>	<u>0.6452</u>	<u>25.24</u>	<u>0.8029</u>
RCAN+ (ours)	$\times 8$	27.47	0.7913	25.40	0.6553	25.05	0.6077	23.22	0.6524	25.58	0.8092

1.2 Results with Bicubic (BI) Degradation Model

We compare our method with 22 state-of-the-art methods: SelfExSR [2], SRCNN [3], FSRCNN [4], SCN [5], VDSR [6], LapSRN [7], DRRN [8], SRRNet [13], SRGAN [13], MemNet [9], SRDenseNet [14], ENet-E [15], ENet-PAT [15], MSLapSRN [16], EDSR [1], MDSR [1], SRMDNF [10], D-DBPN [11], RDN [12], EDSR+ [1], MDSR+ [1] and RDN+ [12]. Similar to [1, 12, 17], we also introduce self-ensemble strategy to further improve our RCAN and denote the self-ensembled one as RCAN+. For fair comparison, we would compare our RCAN with other methods without self-ensemble and RCAN+ with other self-ensembled methods respectively.

Quantitative results without self-ensemble. Table 1 shows quantitative comparisons for $2\times$, $3\times$, $4\times$, and $8\times$ SR. When compared with all previous methods, our RCAN performs the best on all the standard datasets with all scaling factors. It should also be known that SRMDNF [10] and D-DBPN [11] not only use 800 DIV2K training images as we use, but also introduce very large number of extra images for training. Even we use much less number of training images (e.g., 800 DIV2K images), our RCAN can still outperform these leading methods. One reason is that the very deep network of our RCAN has stronger representational property.

Quantitative results with self-ensemble. Table 2 shows quantitative comparisons using self-ensemble [1, 12, 17]. We also add RCAN in the table to show its superior performance. As we can see that RCAN+ performs the best on all the datasets with all scaling factors. Even without self-ensemble, our RCAN still performs the second best in several cases, where our RCAN outperforms

other self-ensembled methods (e.g., EDSR+, MDSR+, and RDN+). These comparisons demonstrate the effectiveness of our proposed RCAN and RCAN+.

Visual comparisons with generative network based methods. In Figure 2, we show the visual comparisons with generative network based methods for $4\times$ SR. We can see that most of previous leading SR methods would suffer from heavy blurring artifacts and cannot recover some details. For example, in images “img_044” and “img_067”, all the compared methods suffer from heavy blurring artifacts. While, our proposed RCAN can alleviate the blurring artifact to some degree and recover more details.

In some other cases (e.g., images “img_012”, “img_092”, and “img_93”), all the compared methods generate wrong details, such as the lines with wrong directions. In contrary, our proposed RCAN can recover lines with right directions.

In addition to the edges and lines above, texture can be more difficult for SR methods to recover. For example, in image “img_076”, all the compared methods generate ruleless textures with some blurring artifacts, failing to recover the textures of bricks. However, our proposed RCAN can reconstruct finer textures, being more faithful to the ground truth.

These comparisons show that previous leading generative network based methods may suffer from heavy blurring artifacts and cannot recover some details. It indicates that these methods have limited ability to handle more challenging cases. On the other hand, our proposed RCAN can alleviate the blurring artifacts to great degree and can recover more details. This superior visual results are not only in consistency with the quantitative results in Table 1, but also demonstrate the better representational ability of our RCAN.

Visual comparisons with GAN-based methods. Recently, with the usage of generative adversarial network (GAN), some image SR methods have been proposed, such as SRGAN [13] and ENet [15]. They claim that their methods can enhance the visual results, even though the quantitative results are relatively low. In Figure 3, we compare our RCAN with GAN-based methods visually.

Specifically, we compare $4\times$ SR results with SRResNet [13], SRResNet_VGG22 [13], SRGAN_MSE [13], SRGAN_VGG22 [13], SRGAN_VGG54 [13], ENet_E [15], and ENet_PAT [15]. SResNet, SRResNet_VGG22, and ENet_E only use generative networks. We also take their visual results into comparisons. The remaining methods use both generative and adversarial networks with different content losses. Readers can refer to [13, 15] for more details. All the results are released by their authors.

We can see from Figure 3 that SRResNet and ENet_E would suffer from heavy blurring artifacts, being similar as those (see Figure 2) by generative network based methods. SRResNet_VGG22 would suffer from checkerboard artifacts. GAN-based methods (e.g., SRGAN_MSE, SRGAN_VGG22, SRGAN_VGG54, and ENet_PAT) would produce unpleasing artifacts. For example, in image “barbara”, all the compared methods fail to recover the book edges. However, our proposed RCAN recovers much better results. In image “253027”, all the compared methods generate heavy artifacts or wrong details for the horse mouth.

While, our proposed RCAN recovers the correct texture of the horse mouth. Similar observations can be found in other images.

These comparisons show that GAN-based methods don't perform well and usually produce unpleasing artifacts. One main reason is that it's hard to control the adversarial training. According to these observations and analyses, we find that SR results with adversarial training may not be faithful to the ground truth. This conclusion is in consistency with the observation in LapSRN [7]. These comparison further demonstrate the effectiveness of our proposed RCAN with more powerful representational ability.

Visual comparisons for 8× SR. In Figure 4, we further provide visual comparisons for 8× SR, a more challenging case in image SR. When the scaling factor is very large, most compared methods would suffer from heavy blurring artifacts and cannot recover details (e.g., see images “img_008”, “img_014”, “img_023”, and “Donburakokko”). Even in such challenging cases, our proposed RCAN can recover more details than all the compared methods. In images “Akkerakanjinchou” and “ShimatteIkouze_vol01”, our RCAN recovers much better results with more clear edges and other details. These comparisons demonstrate that our RCAN can obtain more useful information and produce finer results.

1.3 Results with Blur-downscale (BD) Degradation Model

We also show more results for 3× SR with BD degradation model.

Quantitative results. As shown in Table 3, our RCAN+ achieves the best performance on each dataset. Even without self-ensemble, our RCAN obtains the second best results in most cases.

Visual results. We further show visual results in Figure 5. As we can see, our proposed RCAN can recover more details than other compared methods. This demonstrates the effectiveness of our RCAN with BD degradation model.

Table 3. Quantitative with BD degradation model. Best and second best results are highlighted and underlined

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM								
Bicubic	×3	28.78	0.8308	26.38	0.7271	26.33	0.6918	23.52	0.6862	25.46	0.8149
SPMSR [18]	×3	32.21	0.9001	28.89	0.8105	28.13	0.7740	25.84	0.7856	29.64	0.9003
SRCNN [3]	×3	32.05	0.8944	28.80	0.8074	28.13	0.7736	25.70	0.7770	29.47	0.8924
FSRCNN [4]	×3	26.23	0.8124	24.44	0.7106	24.86	0.6832	22.04	0.6745	23.04	0.7927
VDSR [6]	×3	33.25	0.9150	29.46	0.8244	28.57	0.7893	26.61	0.8136	31.06	0.9234
IRCNN [19]	×3	33.38	0.9182	29.63	0.8281	28.65	0.7922	26.77	0.8154	31.15	0.9245
SRMDNF [10]	×3	34.01	0.9242	30.11	0.8364	28.98	0.8009	27.50	0.8370	32.97	0.9391
RDN [12]	×3	34.58	0.9280	30.53	0.8447	29.23	0.8079	28.46	0.8582	33.97	0.9465
RDN+ [12]	×3	<u>34.70</u>	<u>0.9289</u>	30.64	<u>0.8463</u>	29.30	0.8093	28.67	0.8612	34.34	0.9483
RCAN (ours)	×3	34.70	0.9288	30.63	0.8462	29.32	<u>0.8093</u>	<u>28.81</u>	0.8647	34.38	0.9483
RCAN+ (ours)	×3	34.83	0.9296	30.76	0.8479	29.39	0.8106	29.04	0.8682	34.76	0.9502

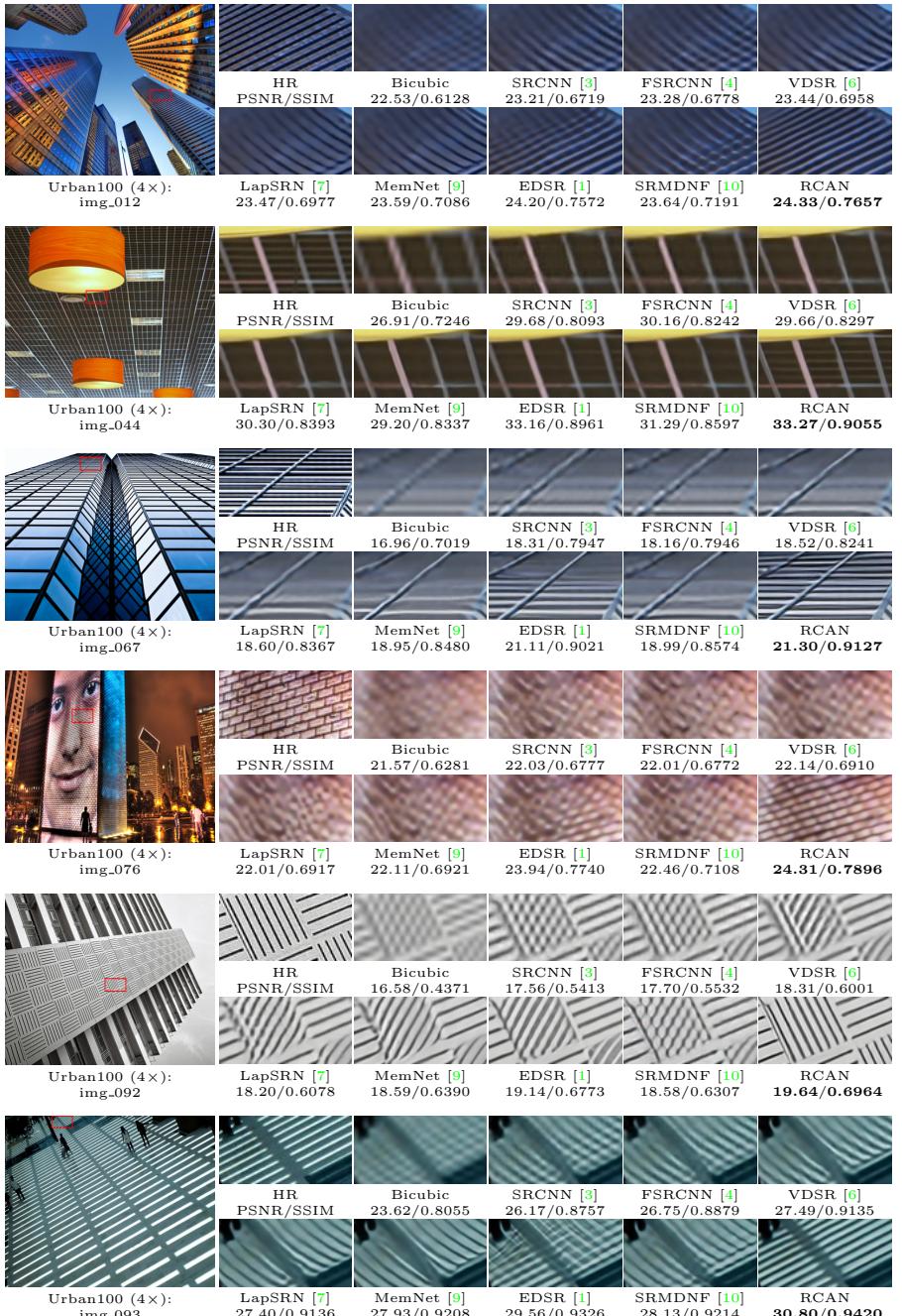


Fig. 2. Visual comparison for 4× SR with BI model on Urban100 dataset. The best results are **highlighted**. These comparisons mainly show the effectiveness of our proposed RCAN against generative network based methods

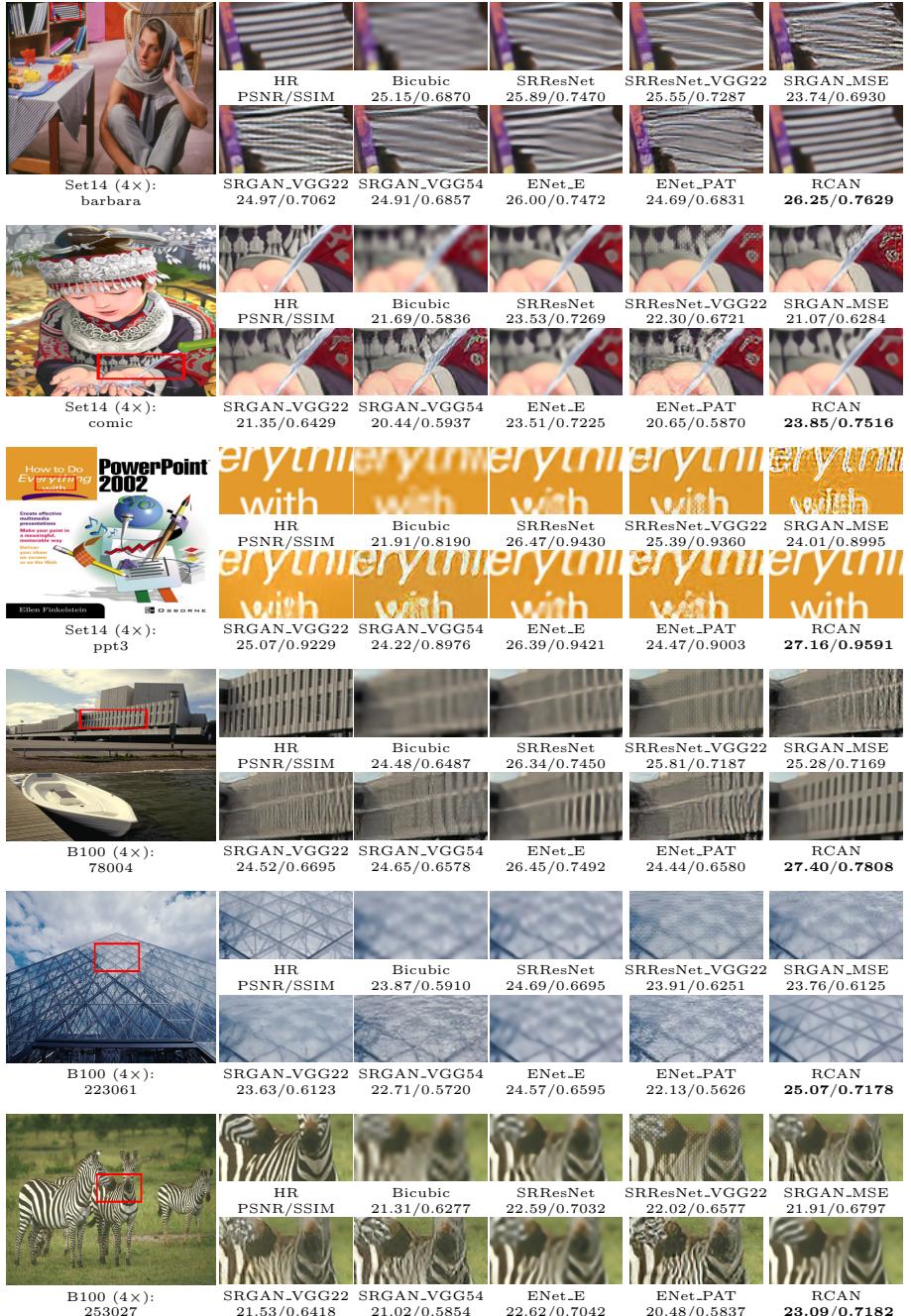


Fig. 3. Visual comparison for 4x SR with BI model on Set14 and B100 datasets. The best results are highlighted. SRResNet, SRResNet_VGG22, SRGAN_MSE, SRGAN_VGG22, and SRGAN_VGG54 are proposed in [13], ENet_E and ENet_PAT are proposed in [15]. These comparisons mainly show the effectiveness of our proposed RCAN against GAN based methods.

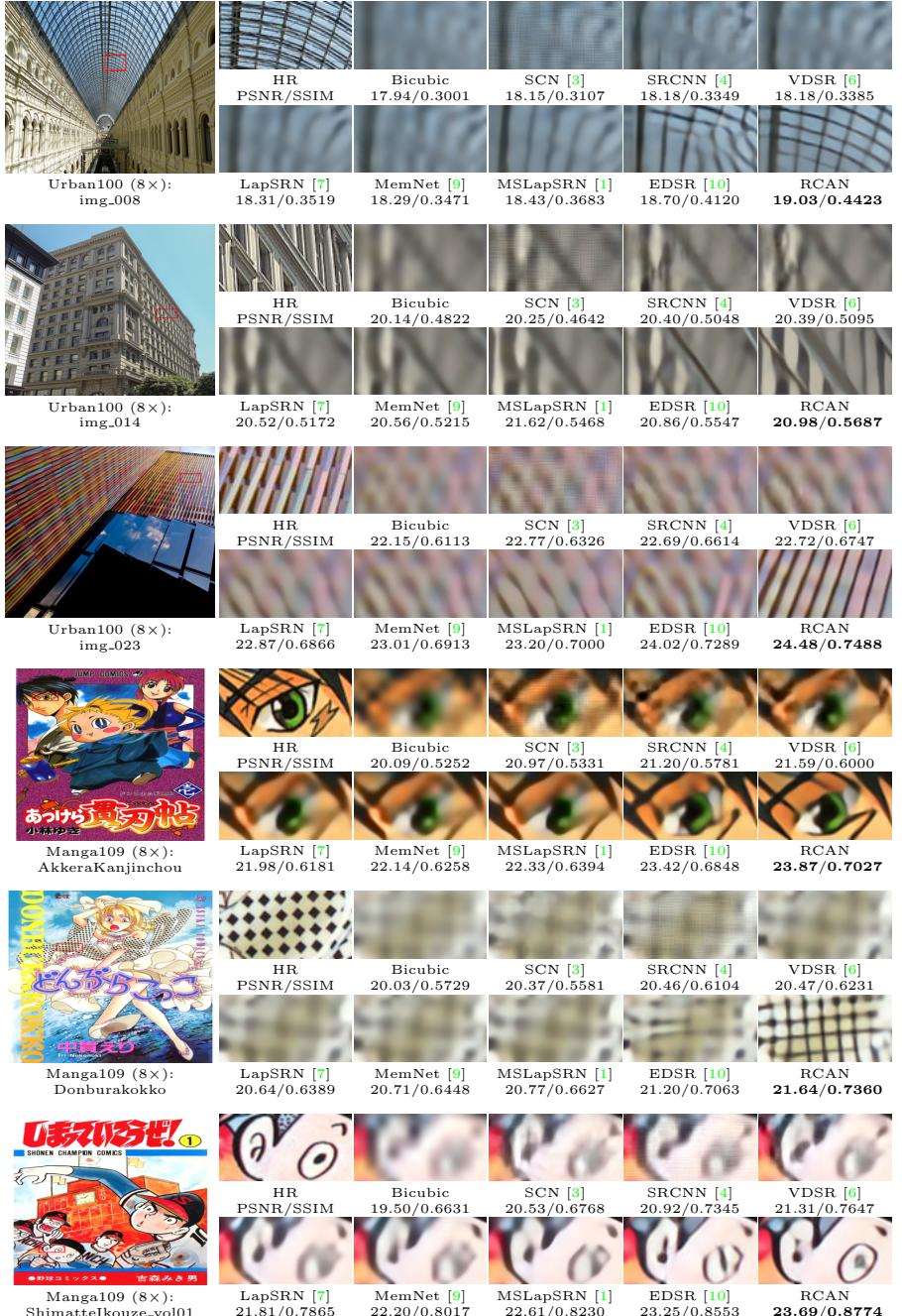


Fig. 4. Visual comparison for 8× SR with BI model on Urban100 and Manga109 datasets. The best results are **highlighted**. These comparisons mainly show the effectiveness of our proposed RCAN against generative network based methods for very large scaling factor (e.g., 8)

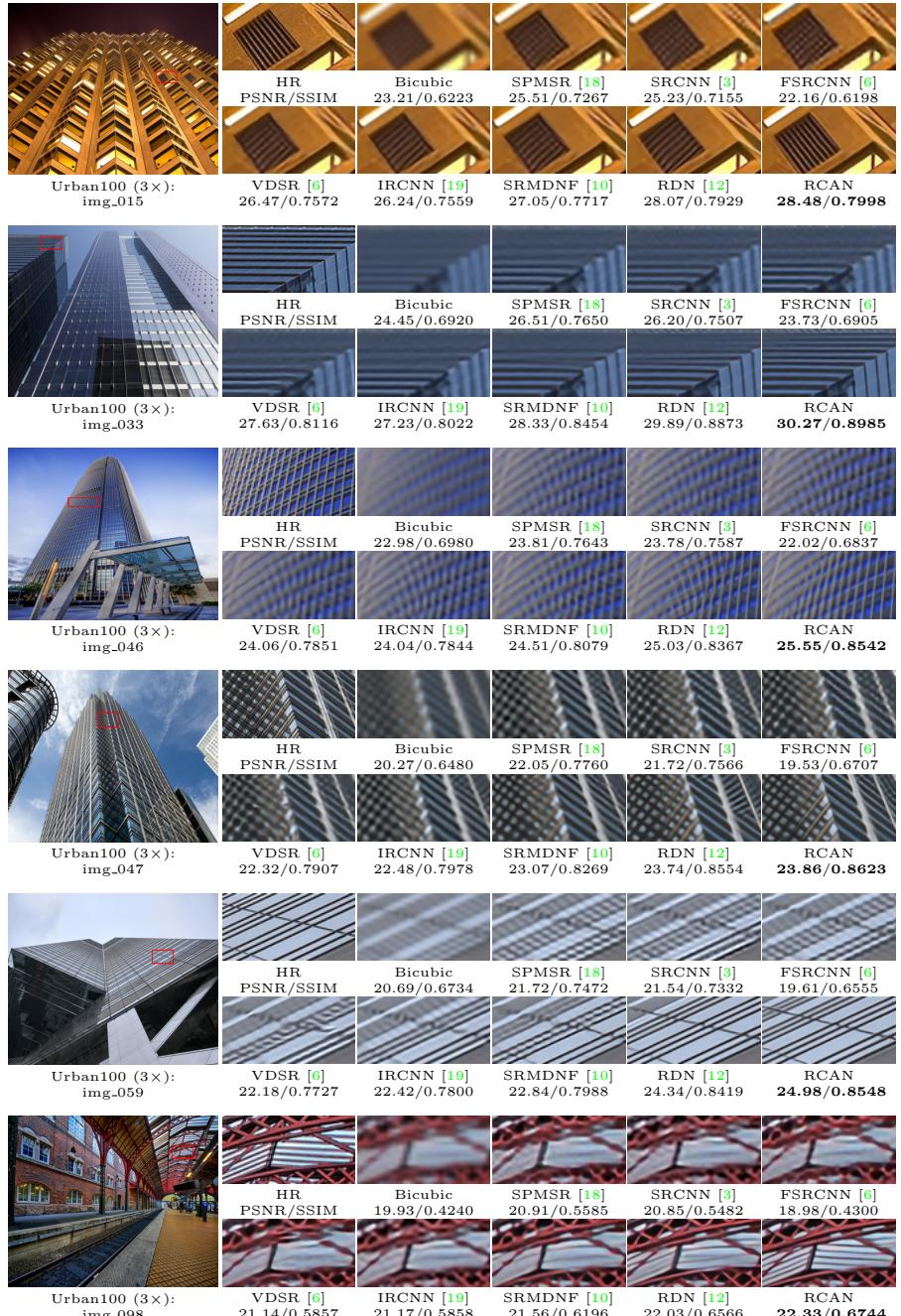


Fig. 5. Visual comparison for 3x SR with BD model on Urban100 dataset. The best results are **highlighted**. These comparisons mainly show the effectiveness of our proposed RCAN against generative network based methods with BD model

References

1. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPRW. (2017)
2. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. (2015)
3. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. TPAMI (2016)
4. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV. (2016)
5. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: ICCV. (2015)
6. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. (2016)
7. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. (2017)
8. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. (2017)
9. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: ICCV. (2017)
10. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: CVPR. (2018)
11. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR. (2018)
12. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. (2018)
13. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. (2017)
14. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: ICCV. (2017)
15. Sajjadi, M.S., Schölkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: ICCV. (2017)
16. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Fast and accurate image super-resolution with deep laplacian pyramid networks. arXiv:1710.01992 (2017)
17. Timofte, R., Rothe, R., Van Gool, L.: Seven ways to improve example-based single image super resolution. In: CVPR. (2016)
18. Peleg, T., Elad, M.: A statistical prediction model based on sparse representations for single image super-resolution. TIP (2014)
19. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: CVPR. (2017)