Toward Real-World Single Image Super-Resolution: A New Benchmark and A New Model - Supplementary Material -

Jianrui Cai[1,*], Hui Zeng[1,*], Hongwei Yong[1,3], Zisheng Cao[2], Lei Zhang[1,3,§]
[1]The Hong Kong Polytechnic University, [2]DJI Co.,Ltd, [3]DAMO Academy, Alibaba Group
{csjcai, cshzeng, cshyong, cslzhang}@comp.polyu.edu.hk, zisheng.cao@dji.com

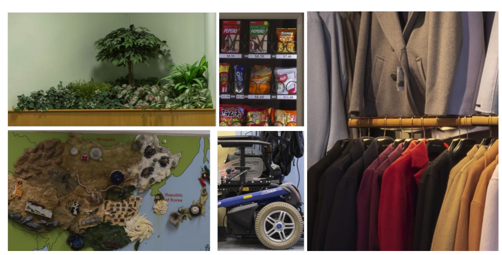
In this supplementary file, we provide:

1. Some sample images of our RealSR dataset.
2. The details of our network architecture.
3. More visual results by SISR models trained on simulated SISR datasets and our RealSR dataset.
4. The computational cost of competing models and their visual results when trained on our RealSR dataset.
5. More super-resolved results on images outside our dataset.

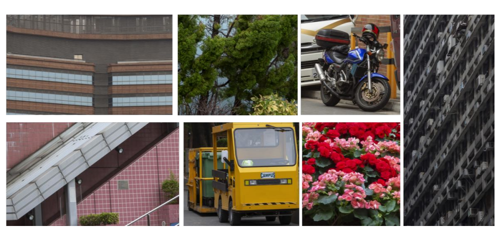
1. Sample images of the RealSR dataset

Currently, the proposed RealSR dataset contains 595 HR-LR image pairs covering a variety of image contents. To ensure the diversity of our RealSR dataset, images are captured in indoor, outdoor and laboratory environments. Several examples of our RealSR dataset are shown in Fig. 1. It provides, to the best of our knowledge, the first general purpose benchmark for real-world SISR model training and evaluation. The RealSR dataset will be made publicly available.

(a) Laboratory scenes



(b) Indoor scenes



(c) Outdoor scenes

Figure 1. Some sample images of our RealSR dataset.

*The first two authors contribute equally to this work.

§Corresponding author. This work is supported by China NSFC grant (no. 61672446) and Hong Kong RGC GRF grant (PolyU 152216/18E).

2. The details of our network architecture

The network architecture of our proposed Laplacian pyramid based kernel prediction network (LP-KPN) is shown in Table 1. In this table, "$H \times W \times C$ conv" denotes a convolutional layer with C filters of size $H \times W$ which is immediately followed by a ReLU nonlinearity. Each residual block contains two 3×3 convolutional layers with the same number of filters on both layers. The stride size for all convolution layers is set to 1 and the number of filters C in each layer is set to 64, except for the last layer where C is set to 25. The structure of the residual block is shown in Fig. 2, which is same as [3]. We use the shuffle operation [4] to downsample and upsample the the image.

Table 1. Network architecture of the proposed LP-KPN.

Layer			Activation size		
Input			192×192		
Shuffle, /4			$48 \times 48 \times 16$		
$3 \times 3 \times 64$ conv, pad 1			$48 \times 48 \times 64$		
$16\times$ Residual blocks, 64 filters			$48 \times 48 \times 64$		
$3 \times 3 \times 64$ conv, pad 1			$48 \times 48 \times 64$		
Shuffle, $\times 4$	Shuffle, $\times 2$	-	$192 \times 192 \times 4$	$96 \times 96 \times 16$	$48 \times 48 \times 64$
$3 \times 3 \times 64$ conv, pad 1	$3 \times 3 \times 64$ conv, pad 1	$3 \times 3 \times 64$ conv, pad 1	$192 \times 192 \times 64$	$96 \times 96 \times 64$	$48 \times 48 \times 64$
$3 \times 3 \times 64$ conv, pad 1	$3 \times 3 \times 64$ conv, pad 1	$3 \times 3 \times 64$ conv, pad 1	$192 \times 192 \times 64$	$96 \times 96 \times 64$	$48 \times 48 \times 64$
$3 \times 3 \times 25$ conv, pad 1	$3 \times 3 \times 25$ conv, pad 1	$3 \times 3 \times 25$ conv, pad 1	$192 \times 192 \times 25$	$96 \times 96 \times 25$	$48 \times 48 \times 25$
Per-pixel conv by Eq. (8)	Per-pixel conv by Eq. (8)	Per-pixel conv by Eq. (8)	192×192	96×96	48×48
Output (Laplacian pyramid reconstruction)			192×192		

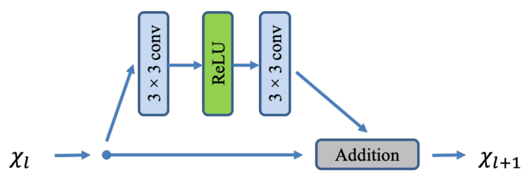


Figure 2. Residual block used in our network.

3. More visual results by SISR models trained on simulated SISR datasets and our RealSR dataset

In this section, we provide more visual results by SISR models trained on simulated SISR datasets (BD and MD [5]) and our proposed RealSR dataset. Two images captured by Canon 5D3, two images captured by Nikon D810 and their super-resolved results are shown in Fig. 3. Again, the models trained on our RealSR dataset consistently obtain better visual quality compared to their counterparts trained on simulated datasets.

4. The computational cost of competing models and their visual results when trained on our RealSR dataset

The running time and the number of parameters of the competing models are listed in Table 2. One can see that although larger kernel size can consistently bring better results for the KPN architecture, the number of parameters will also greatly increase. Benefitting from the Laplacian pyramid decomposition strategy, our LP-KPN using 5×5 kernel can achieve better results than the KPN using 19×19 kernel, and it uses much less parameters. Specifically, our LP-KPN model contains less than $\frac{1}{5}$ parameters of the RCAN model [6] and it runs about 3 times faster than RCAN. The visual examples of the SISR results by the competing models are shown in Fig. 4. Though all the SISR models in Fig. 4 are trained on our RealSR dataset and they all achieve good results, our LP-KPN still obtains the best visual quality among the competitors.

5. More super-resolved results on images outside our dataset

In this section, we provide more super-resolved results on images outside our dataset, including images taken by one Sony a7II DSLR camera and two mobile cameras (*i.e.*, iPhone X and Google Pixel 2). The visual examples are shown in Fig. 5.

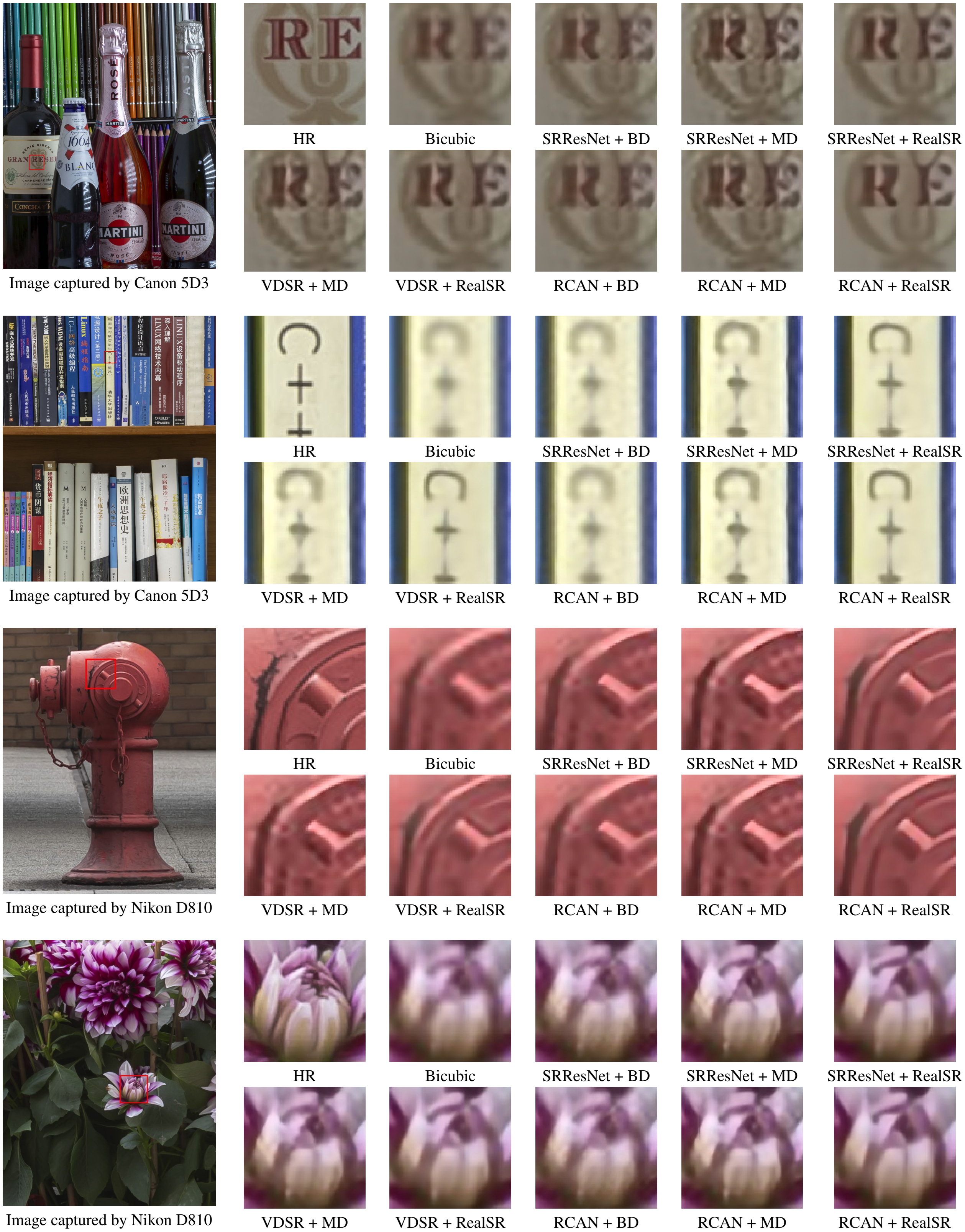


Figure 3. SR results ($\times 4$) on our RealSR testing set by different methods (trained on different datasets).

Table 2. PSNR, SSIM, running time and parameters for different models (trained on our RealSR training set) on our RealSR testing set. The running time is measured for an image of size 1200×2200. We use the file size of Caffe model to represent the number of parameters.

		Bicubic	VDSR [1]	SRResNet [2]	RCAN [6]	DPS	KPN, k = 5	KPN, k = 7	KPN, k = 13	KPN, k = 19	Our, k = 5
PSNR	$\times 2$	32.61	33.64	33.69	33.87	33.71	33.75	33.78	33.83	33.86	33.90
	$\times 3$	29.34	30.14	30.18	30.40	30.20	30.26	30.29	30.35	30.39	30.42
	$\times 4$	27.99	28.63	28.67	28.88	28.69	28.74	28.78	28.85	28.90	28.92
SSIM	$\times 2$	0.907	0.917	0.919	0.922	0.919	0.920	0.921	0.923	0.924	0.927
	$\times 3$	0.841	0.856	0.859	0.862	0.859	0.860	0.861	0.862	0.864	0.868
	$\times 4$	0.806	0.821	0.824	0.826	0.824	0.826	0.827	0.828	0.830	0.834
Parameters		-	2.667M	5.225M	32.71M	5.079M	5.134M	5.190M	5.467M	5.910M	5.731M
Times (sec.)		-	0.4262	0.1431	0.5106	0.1228	0.1296	0.1391	0.1909	0.2748	0.1813

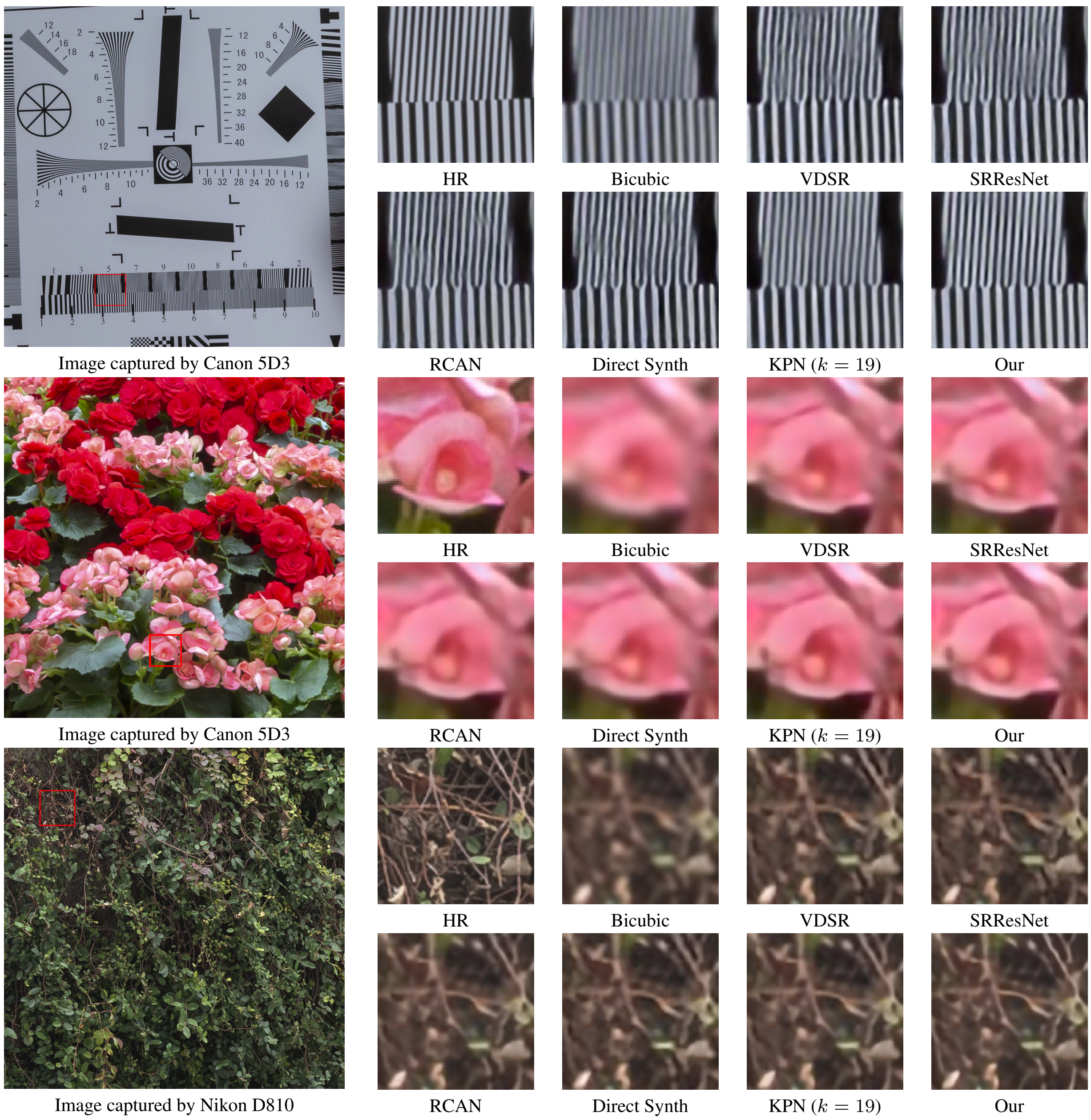


Figure 4. SR results ($\times 3$) on our RealSR testing set by different methods (all trained on our RealSR dataset). It can be seen that all SISR models trained on our RealSR dataset achieve good results, while our LP-KPN still obtains the best visual quality.

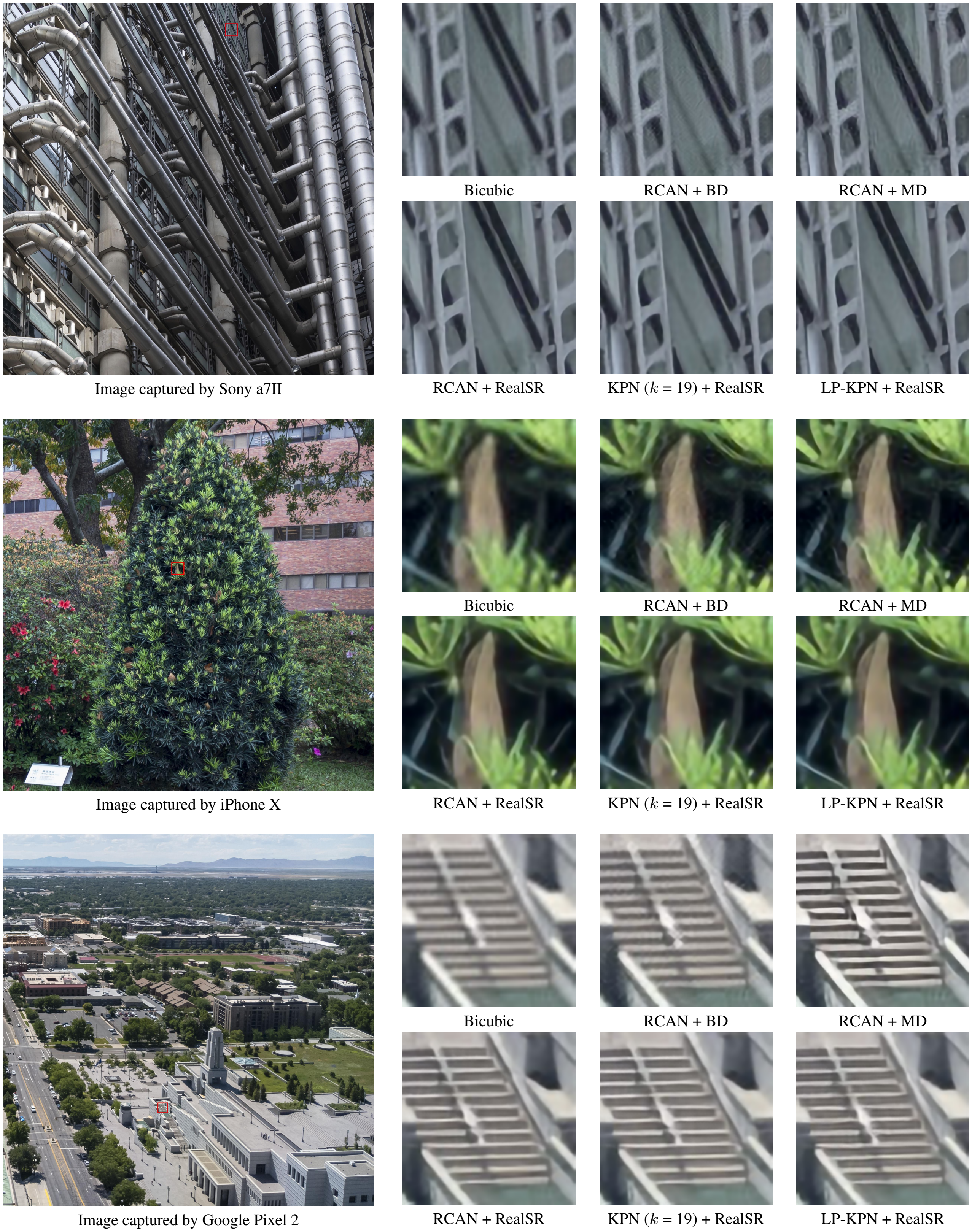


Figure 5. SISR results ($\times 4$) of real-world images outside our dataset. Images are captured by Sony a7II, iPhone X and Google Pixel 2.

References

[1] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 4

[2] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 4

[3] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017. 2

[4] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 2

[5] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*. 2

[6] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 2, 4