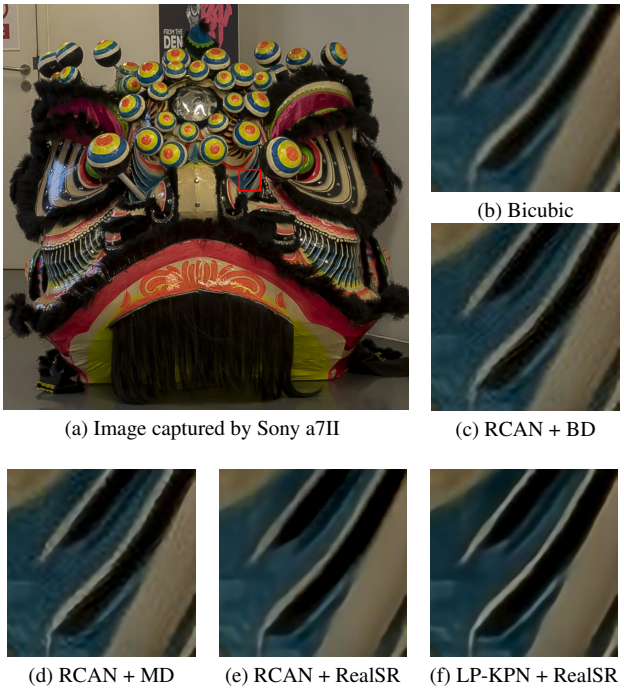
Toward Real-World Single Image Super-Resolution: A New Benchmark and A New Model

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

Abstract

Most of the existing learning-based single image super-resolution (SISR) methods are trained and evaluated on simulated datasets, where the low-resolution (LR) images are generated by applying a simple and uniform degradation (i.e., bicubic downsampling) to their high-resolution (HR) counterparts. However, the degradations in real-world LR images are far more complicated. As a consequence, the SISR models trained on simulated data become less effective when applied to practical scenarios. In this paper, we build a real-world super-resolution (RealSR) dataset where paired LR-HR images on the same scene are captured by adjusting the focal length of a digital camera. An image registration algorithm is developed to progressively align the image pairs at different resolutions. Considering that the degradation kernels are naturally non-uniform in our dataset, we present a Laplacian pyramid based kernel prediction network (LP-KPN), which efficiently learns per-pixel kernels to recover the HR image. Our extensive experiments demonstrate that SISR models trained on our RealSR dataset deliver better visual quality with sharper edges and finer textures on real-world scenes than those trained on simulated datasets. Though our RealSR dataset is built by using only two cameras (Canon 5D3 and Nikon D810), the trained model generalizes well to other camera devices such as Sony a7II and mobile phones.



(a) Image captured by Sony a7II (b) Bicubic (c) RCAN + BD (d) RCAN + MD (e) RCAN + RealSR (f) LP-KPN + RealSR

Figure 1. The SISR results ($\times 4$) of (a) a real-world image captured by a Sony a7II camera. SISR results generated by (b) bicubic interpolator, RCAN models [64] trained using image pairs (in DIV2K [46]) with (c) bicubic degradation (BD), (d) multiple simulated degradations (MD) [62], and (e) authentic distortions in our RealSR dataset. (f) SISR result by the proposed LP-KPN model trained on our dataset. Note that our RealSR dataset is collected by Canon 5D3 and Nikon D810 cameras.

1. Introduction

Single image super-resolution (SISR) [16] aims to recover a high-resolution (HR) image from its low-resolution (LR) observation. SISR has been an active research topic for decades [39, 59, 46, 48, 4, 6] because of its high practical values in enhancing image details and textures. Since SISR is a severely ill-posed inverse problem, learning image prior information from HR and/or LR exemplar images [16, 14, 57, 20, 15, 8, 25, 58, 12, 21, 47, 42] plays an indispensable role in recovering details from an LR image. Benefitting from the rapid development of deep convolutional neural networks (CNNs) [29], recent years have witnessed an explosive spread of training CNN models to perform SISR, and the performance has been consistently improved by designing new CNN architectures [10, 51, 43, 24, 45, 31, 65, 64] and loss functions [23, 30, 41].

*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).

Though significant advances have been made, most of the existing SISR methods are trained and evaluated on simulated datasets which assume simple and uniform degradation (*i.e.*, bicubic degradation). Unfortunately, SISR models trained on such simulated datasets are hard to generalize to practical applications since the authentic degradations in real-world LR images are much more complex [56, 27]. Fig. 1 shows the SISR results of a real-world image captured by a Sony a7II camera. We utilize the state-of-the-art RCAN method [64] to train three SISR models using simulated image pairs (in DIV2K [46]) with bicubic degradation, multiple simulated degradations [62] and image pairs with authentic distortions in our dataset to be constructed in this paper. The results clearly show that, compared with the simple bicubic interpolator (Fig. 1(b)), the RCAN models trained on simulated datasets (Figs. 1(c)∼1(d)) do not show clear advantages on real-world images.

It is thus highly desired that we can have a training dataset consisting of real-world, instead of simulated, LR and HR image pairs. However, constructing such a real-world super-resolution (RealSR) dataset is a non-trivial job since the ground-truth HR images are very difficult to obtain. In this work, we aim to construct a general and practical RealSR dataset using a flexible and easy-to-reproduce method. Specifically, we capture images of the same scene using fixed digital single-lens reflex (DSLR) cameras with different focal lengths. By increasing the focal length, finer details of the scene can be naturally recorded into the camera sensor. In this way, HR and LR image pairs on different scales can be collected. However, in addition to the change of field of view (FoV), adjusting focal length can result in many other changes in the imaging process, such as shift of optical center, variation of scaling factors, different exposure time and lens distortion. We thus develop an effective image registration algorithm to progressively align the image pairs such that the end-to-end training of SISR models can be performed. The constructed RealSR dataset contains various indoor and outdoor scenes taken by two DSLR cameras (Canon 5D3 and Nikon D810), providing a good benchmark for training and evaluating SISR algorithms in practical applications.

Compared with the previous simulated datasets, the image degradation process in our RealSR dataset is much more complicated. In particular, the degradation is spatially variant since the blur kernel varies with the depth of content in a scene. This motivates us to train a kernel prediction network (KPN) for the real-world SISR task. The idea of kernel prediction is to explicitly learn a restoration kernel for each pixel, and it has been employed in applications such as denoising [1, 35, 49], dynamic deblurring [44, 17] and video interpolation [36, 37]. Though effective, the memory and computational cost of KPN is quadratically increased with the kernel size. To obtain as competitive SISR performance as using large kernel size while achieving high computational efficiency, we propose a Laplacian pyramid based KPN (LP-KPN) which learns per-pixel kernels for the decomposed image pyramid. Our LP-KPN can leverage rich information using a small kernel size, leading to effective and efficient real-world SISR performance. Figs. 1(e) and 1(f) show the SISR results of RCAN [64] and LP-KPN models trained on our RealSR dataset, respectively. One can see that both of them deliver much better results than the RCAN models trained on simulated data, while our LP-KPN (46 conv layers) can output more distinct result than RCAN (over 400 conv layers) using much fewer layers.

The contributions of this work are twofold:

- We build a RealSR dataset consisting of precisely aligned HR and LR image pairs with different scaling factors, providing a general purpose benchmark for real-world SISR model training and evaluation.
- We present an LP-KPN model and validate its efficiency and effectiveness in real-world SISR.

Extensive experiments are conducted to quantitatively and qualitatively analyze the performance of our RealSR dataset in training SISR models. Though the dataset in its current version is built using only two cameras, the trained SISR models exhibit good generalization capability to images captured by other types of camera devices.

2. Related Work

SISR datasets. There are several popular datasets, including Set5 [3], Set14 [61], BSD300 [33], Urban100 [20], Manga109 [34] and DIV2K [46] that have been widely used for training and evaluating the SISR methods. In all these datasets, the LR images are generally synthesized by a simple and uniform degradation process such as bicubic downsampling or Gaussian blurring followed by direct downsampling [11]. The SISR Models trained on these simulated data may exhibit poor performance when applied to real LR images where the degradation deviates from the simulated ones [13]. To improve the generalization capability, Zhang *et al*. [62] trained their model using multiple simulated degradations and Bulat *et al*. [5] used a GAN [18] to generate the degradation process. Although these more advanced methods can simulate more complex degradation, there is no guarantee that such simulated degradation can approximate the authentic degradation in practical scenarios which is usually very complicated [27].

Several recent attempts have been made on capturing real-world image pairs for SISR. Qu *et al*. [40] put two cameras together with a beam splitter to collect a dataset with paired face images. Köhler *et al*. [27] employed hardware binning on the sensor to capture LR images and used multiple postprocessing steps to generate different versions of an LR image. However, both datasets were collected in

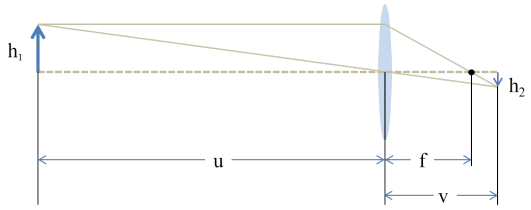


Figure 2. Illustration of thin lens. u, v, f represent the object distance, image distance and focal length, respectively. h_1 and h_2 denote the size of object and image.

indoor laboratory environment and very limited number of scenes (31 face images in [40] and 14 scenes in [27]) were included. More recently, two contemporary datasets have been constructed using similar strategy as ours. Chen *et al*. [7] captured 100 image pairs of printed postcards at one scaling factor, but the models trained on this dataset may not generalize well to real-world natural scenes. Zhang *et al*. [63] captured 500 scenes using multiple focal lengths. However, the image pairs are not precisely aligned in this dataset, making it inconvenient to evaluate the performance of trained models on this dataset. Different from them, in our dataset we captured images from various scenes at multiple focal lengths, and developed a systematic image registration algorithm to precisely align the image pairs, providing a general and easy-to-use benchmark for real-world single image super-resolution.

Kernel prediction networks. Considering that the degradation kernel in our RealSR dataset is spatially variant, we propose to train a kernel prediction network (KPN) for real-world SISR. The idea of KPN was first proposed in [1] to denoise Monte Carlo renderings and it has proven to have faster convergence and better stability than direct prediction [49]. Mildenhall *et al*. [35] trained a KPN model for burst denoising and obtained state-of-the-art performance on both synthetic and real data. Similar ideas have been employed in estimating the blur kernels in dynamic deblurring [44, 17] or convolutional kernels in video interpolation [36, 37]. We are among the first to train a KPN for SISR and we propose the LP-KPN to perform kernel prediction in the scale space with high efficiency.

3. Real-world SISR Dataset

To build a dataset for learning and evaluating real-world SISR models, we propose to collect images of the same scene by adjusting the lens of DSLR cameras. Sophisticated image registration operations are then performed to generate the HR and LR pairs of the same content. The detailed dataset construction process is presented in this section.

3.1. Image formation by thin lens

The DSLR camera imaging system can be approximated as a thin lens [54]. An illustration of the image formation process by thin lens is shown in Fig. 2. We denote the object distance, image distance and focal length by u, v, f, and denote the size of object and image by h_1 and h_2, respectively. The lens equation is defined as follows [54]:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}. \tag{1}$$

The magnification factor M is defined as the ratio of the image size to the object size:

$$M = \frac{h_2}{h_1} = \frac{v}{u}. \tag{2}$$

In our case, the static images are taken at a distance (*i.e*., u) larger than 3.0m. Both h_1 and u are fixed and u is much larger than f (the largest f is 105mm). Combining Eq. (1) and Eq. (2), and considering the fact that $u \gg f$, we have:

$$h_2 = \frac{f}{u - f} h_1 \approx \frac{f}{u} h_1. \tag{3}$$

Therefore, h_2 is approximately linear to f. By increasing the focal length f, larger images with finer details will be recorded in the camera sensor. The scaling factor can also be controlled (in theory) by choosing specific values of f.

Table 1. Number of image pairs for each camera at each scaling factor.

Camera	Canon 5D3			Nikon D810		
Scale	×2	×3	×4	×2	×3	×4
# image pairs	86	117	86	97	117	92

3.2. Data collection

We used two full frame DSLR cameras (Canon 5D3 and Nikon D810) to capture images for data collection. The resolution of Canon 5D3 is 5760×3840, and that of Nikon D810 is 7360×4912. To cover the common scaling factors (*e.g*., ×2, ×3, ×4) used in most previous SISR datasets, both cameras were equipped with one 24∼105mm, f/4.0 zoom lens. For each scene, we took photos using four focal lengths: 105mm, 50mm, 35mm, and 28mm. Images taken by the largest focal length are used to generate the ground-truth HR images, and images taken by the other three focal lengths are used to generate the LR versions. We choose 28mm rather than 24mm because lens distortion at 24mm is more difficult to correct in post-processing, which results in less satisfied quality in image pair registration.

The camera was set to aperture priority mode and the aperture was adjusted according to the depth-of-field (DoF) [53]. Basically, the selected aperture value should make the DoF large enough to cover the scene and avoid severe diffraction. Small ISO is preferred to alleviate noise. The focus, white balance, and exposure were set to automatic mode. The center-weighted metering option was selected since only the center region of captured images were used

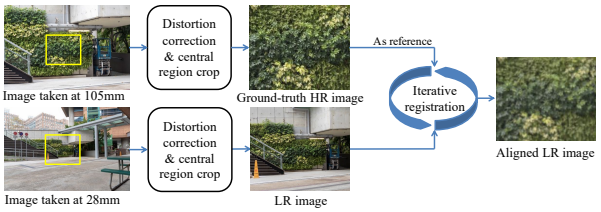


Figure 3. Illustration of our image pair registration process.

in our final dataset. For stabilization, the camera was fixed on a tripod and a bluetooth remote controller was used to control the shutter. Besides, lens stabilization was turned off and the reflector was pre-rised when taking photos.

To ensure the generality of our dataset, we took photos in both indoor and outdoor environment. Scenes with abundant texture are preferred considering that the main purpose of super-resolution is to recover or enhance image details. For each scene, we first captured the image at 105mm focal length and then manually decreased the focal length to take three down-scaled versions. 234 scenes were captured, and there are no overlapped scenes between the two cameras. After discarding images having moving objects, inappropriate exposure, and blur, we have 595 HR and LR image pairs in total. The numbers of image pairs for each camera at each scaling factor are listed in Table 1.

3.3. Image pair registration

Although it is easy to collect images on different scales by zooming the lens of a DSLR camera, it is difficult to obtain pixel-wise aligned image pairs because the zooming of lens brings many uncontrollable changes. Specifically, images taken at different focal lengths suffer from different lens distortions and usually have different exposures. Moreover, the optical center will also shift when zooming the focal length because of the inherent defect of lens [55]. Even the scaling factors are varying slightly because the lens equation (Eq. (1)) cannot be precisely satisfied in practical focusing process. With the above factors, none of the existing image registration algorithms can be directly used to obtain accurate pixel-wise registration of two images captured under different focal length. We thus develop an image registration algorithm to progressively align such image pairs to build our RealSR dataset.

The registration process is illustrated in Fig. 3. We first import the images with meta information into PhotoShop to correct the lens distortion. However, this step cannot perfectly correct the lens distortion especially for the region distant from the optical center. We thus further crop the interested region around the center of the image, where distortion is not severe and can be well corrected. The cropped region from the image taken at 105mm focal length is used as the ground-truth HR image, whose LR counterparts are to be registered from images taken at 50mm, 35mm, or 28mm focal length. Due to the large difference of resolution and small changes in luminance between images taken at different focal lengths, those sparse keypoint based image registration algorithms such as SURF [2] and SIFT [32] cannot always achieve pixel-wise registration, which is necessary for our dataset. To obtain accurate image pair registration, we develop a pixel-wise registration algorithm which simultaneously considers luminance adjustment. Denote by $\mathbf{I}_H$ and $\mathbf{I}_L$ the HR image and the LR image to be registered, our algorithm minimizes the following objective function:

$$\min_{\boldsymbol{\tau}} ||\alpha C(\boldsymbol{\tau} \circ \mathbf{I}_L) + \beta - \mathbf{I}_H||_p^p, \tag{4}$$

where $\boldsymbol{\tau}$ is an affine transformation matrix, C is a cropping operation which makes the transformed $\mathbf{I}_L$ have the same size as $\mathbf{I}_H$, α and β are luminance adjustment parameters, $||\cdot||_p$ is a robust L_p-norm ($p \leq 1$), *e.g.*, L_1-norm.

The above objective function is solved in an iterative manner. At the beginning, according to Eq. (3), the $\boldsymbol{\tau}$ is initialized as a scaling transformation with scaling factor calculated as the ratio of two focal lengths. Let $\mathbf{I}'_L = C(\boldsymbol{\tau} \circ \mathbf{I}_L)$. With $\mathbf{I}'_L$ and $\mathbf{I}_H$ fixed, the parameters for luminance adjustment can be obtained by $\alpha = \text{std}(\mathbf{I}_H)/\text{std}(\mathbf{I}'_L)$ and $\beta = \text{mean}(\mathbf{I}_H) - \alpha\text{mean}(\mathbf{I}'_L)$, which can ensure $\mathbf{I}'_L$ having the same pixel mean and variance as $\mathbf{I}_H$ after luminance adjustment. Then we solve the affine transformation matrix $\boldsymbol{\tau}$ with α and β fixed. According to [38, 60], the objective function w.r.t. $\boldsymbol{\tau}$ is nonlinear, which can be iteratively solved by a locally linear approximation:

$$\min_{\Delta\boldsymbol{\tau}} ||\alpha C(\boldsymbol{\tau} \circ \mathbf{I}_L) + \beta + \alpha\mathbf{J}\Delta\boldsymbol{\tau} - \mathbf{I}_H||_p^p, \tag{5}$$

where $\mathbf{J}$ is the Jacobian matrix of $C(\boldsymbol{\tau} \circ \mathbf{I}_L)$ w.r.t. $\boldsymbol{\tau}$, and this objective function can be solved by an iteratively reweighted least square problem (IRLS) as follows [9]:

$$\min_{\Delta\boldsymbol{\tau}} ||\mathbf{w} \odot (\mathbf{A}\Delta\boldsymbol{\tau} - \mathbf{b})||_2^2, \tag{6}$$

where $\mathbf{A} = \alpha\mathbf{J}$, $\mathbf{b} = \mathbf{I}_H - (\alpha C(\boldsymbol{\tau} \circ \mathbf{I}_L) + \beta)$, $\mathbf{w}$ is the weight matrix and $\odot$ denotes element-wise multiplication. Then we can obtain:

$$\Delta\boldsymbol{\tau} = (\mathbf{A}'\text{diag}(\mathbf{w})^2\mathbf{A})^{-1}\mathbf{A}'\text{diag}(\mathbf{w})^2\mathbf{b}, \tag{7}$$

and $\boldsymbol{\tau}$ can be updated by: $\boldsymbol{\tau} = \boldsymbol{\tau} + \Delta\boldsymbol{\tau}$.

We iteratively estimate the luminance adjustment parameters and the affine transformation matrix. The optimization process converges within 5 iterations since our prior information of the scaling factor provides a good initialization of $\boldsymbol{\tau}$. After convergence, we can obtain the aligned LR image as $\mathbf{I}_L^A = \alpha C(\boldsymbol{\tau} \circ \mathbf{I}_L) + \beta$.

4. Laplacian Pyramid based Kernel Prediction Network

In Section 3, we have constructed a new real-world super-resolution (RealSR) dataset, which consists of pixel-

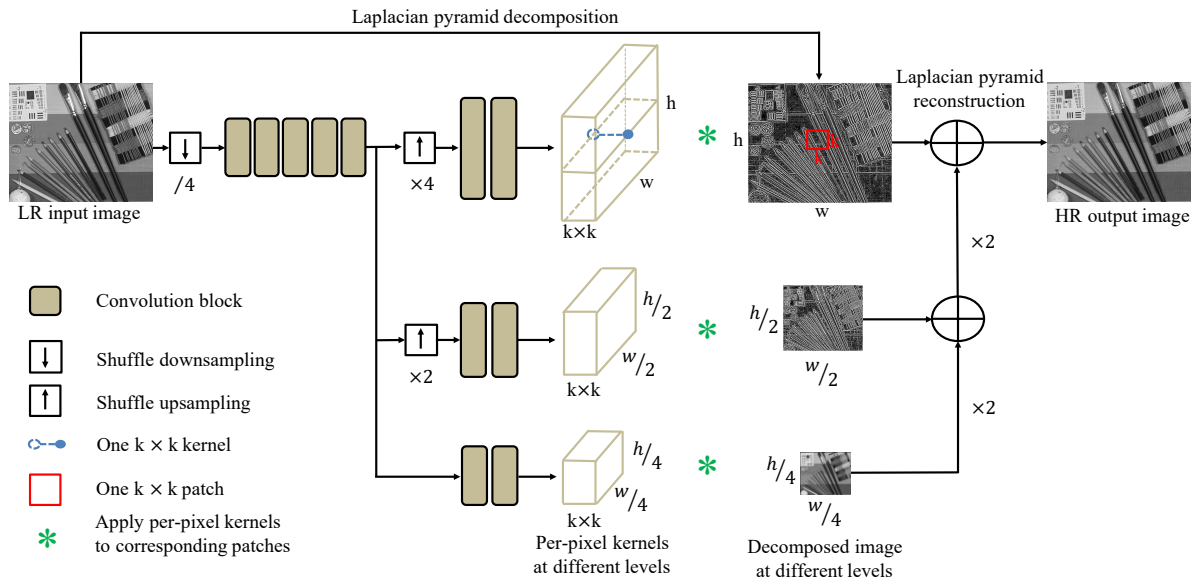


Figure 4. Framework of the Laplacian pyramid based kernel prediction network. By decomposing the image into a Laplacian pyramid, using small kernels can leverage rich neighborhood information for super-resolution.

wise aligned HR and LR image pairs $\{\mathbf{I}_H, \mathbf{I}_L^A\}$ of size $h \times w$. Now the problem turns to how to learn an effective network to enhance $\mathbf{I}_L^A$ to $\mathbf{I}_H$. For LR images in our RealSR dataset, the blur kernel varies with the depth in a scene [52] and the DoF [53] changes with the focal length. Training an SISR model which directly transforms the LR image to the HR image, as done in most of the previous CNN based SISR methods, may not be the cost-effective way. We therefore propose to train a kernel prediction network (KPN) which explicitly learns an individual kernel for each pixel. Compared with those direct pixel synthesis networks, KPN has proven to have advantages in efficiency, interpretability and generalization capability in tasks of denoising, dynamic deblurring, etc., [1, 35, 49, 44, 17, 28].

The KPN takes the $\mathbf{I}_L^A$ as input and outputs a kernel tensor $\mathbf{T} \in R^{(k\times k)\times h\times w}$, in which each vector in channel dimension $\mathbf{T}(i,j) \in R^{(k\times k)}$ can be reshaped into a $k \times k$ kernel $\mathbf{K}(i,j)$. The reshaped per-pixel kernel $\mathbf{K}(i,j)$ is applied to the $k \times k$ neighborhood of each pixel in the input LR image $\mathbf{I}_L^A(i,j)$ to reproduce the HR output. The predicted HR image, denoted by $\mathbf{I}_H^P$, is obtained by:

$$\mathbf{I}_H^P(i,j) = \langle \mathbf{K}(i,j), V(\mathbf{I}_L^A(i,j)) \rangle, \tag{8}$$

where $V(\mathbf{I}_L^A(i,j))$ represents a $k \times k$ neighborhood of pixel $\mathbf{I}_L^A(i,j)$ and $\langle \cdot \rangle$ denotes the inner product operation.

Eq. (8) shows that the output pixel is a weighted linear combination of the neighboring pixels in the input image. To obtain good performance, a large kernel size is necessary to leverage richer neighborhood information, especially when only a single frame image is used. On the other hand, the predicted kernel tensor $\mathbf{T}$ grows quadratically with the kernel size k, which can result in high computational and memory cost in practical applications. In order to train a both effective and efficient KPN, we propose a Laplacian pyramid based KPN (LP-KPN).

The framework of our LP-KPN is shown in Fig. 4. As in many SR methods [31, 48], our model works on the Y channel of YCbCr space. The Laplacian pyramid decomposes an image into several levels of sub-images with downsampled resolution and the decomposed images can exactly reconstruct the original image. Using this property, the Y channel of an LR input image $\mathbf{I}_L^A$ is decomposed into a three-level image pyramid $\{\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2\}$, where $\mathbf{S}_0 \in R^{h\times w}$, $\mathbf{S}_1 \in R^{\frac{h}{2}\times\frac{w}{2}}$, and $\mathbf{S}_2 \in R^{\frac{h}{4}\times\frac{w}{4}}$. Our LP-KPN takes the LR image as input and predicts three kernel tensors $\{\mathbf{T}_0, \mathbf{T}_1, \mathbf{T}_2\}$ for the image pyramid, where $\mathbf{T}_0 \in R^{(k\times k)\times h\times w}$, $\mathbf{T}_1 \in R^{(k\times k)\times\frac{h}{2}\times\frac{w}{2}}$, and $\mathbf{T}_2 \in R^{(k\times k)\times\frac{h}{4}\times\frac{w}{4}}$. The learned kernel tensors $\{\mathbf{T}_0, \mathbf{T}_1, \mathbf{T}_2\}$ are applied to the corresponding image pyramid $\{\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2\}$, using the operation in Eq. (8), to restore the Laplacian decomposition of HR image at each level. Finally, the Laplacian pyramid reconstruction is conducted to obtain the HR image. Benefitting from the Laplacian pyramid, learning three $k \times k$ kernels can equally lead to a receptive field with size $4k \times 4k$ at the original resolution, which significantly reduces the computational cost compared to directly learning one $4k \times 4k$ kernel.

The backbone of our LP-KPN consists of 17 residual blocks, with each residual block containing 2 convolutional layers and a ReLU function (similar structure to [31]). To improve the efficiency, we shuffle [43] the input LR image with factor $\frac{1}{4}$ (namely, the $h \times w$ image is shuffled to 16 $\frac{h}{4} \times \frac{w}{4}$ images) and input the shuffled images to the network. Most convolutional blocks are shared by the three

Table 2. Average PSNR (dB) and SSIM indices on our RealSR testing set by different methods (trained on different datasets).

Metric	Scale	Bicubic	VDSR [24]			SRResNet [30]			RCAN [64]		
			BD	MD	Our	BD	MD	Our	BD	MD	Our
PSNR	×2	32.61	32.63	32.65	33.64	32.66	32.69	33.69	32.91	32.92	33.87
	×3	29.34	29.40	29.43	30.14	29.46	29.47	30.18	29.66	29.69	30.40
	×4	27.99	28.03	28.06	28.63	28.09	28.12	28.67	28.28	28.31	28.88
SSIM	×2	0.907	0.907	0.908	0.917	0.908	0.909	0.919	0.910	0.912	0.922
	×3	0.841	0.842	0.845	0.856	0.844	0.846	0.859	0.847	0.851	0.862
	×4	0.806	0.806	0.807	0.821	0.806	0.808	0.824	0.811	0.813	0.826

levels of kernels except for the last few layers. One $\times 4$ and one $\times 2$ shuffle operation are performed to upsample the spatial resolution of the latent image representations at two lower levels, followed by individual convolutional blocks. Our LP-KPN has a total of 46 convolutional layers, which is much less than the previous state-of-the-art SISR models [31, 65, 64]. The detailed network architecture can be found in the **supplementary material**. The L_2-norm loss function $\mathcal{L}(\mathbf{I}_H, \mathbf{I}_H^P) = ||\mathbf{I}_H - \mathbf{I}_H^P||_2^2$ is employed to minimize the pixel-wise distance between the model prediction $\mathbf{I}_H^P$ and the ground-truth HR image $\mathbf{I}_H$.

5. Experiments

Experimental setup. The number of image pairs in our RealSR dataset is reported in Table 1. We randomly selected 15 image pairs at each scaling factor for each camera to form the testing set, while using the remaining image pairs as training set. Except for cross-camera testing, images from both the Canon and Nikon cameras were combined for training and testing. Following the previous work [31, 64, 48], the SISR results were evaluated using PSNR and SSIM [50] indices on the Y channel in the YCbCr space. The height and width of images lie in the range of [700, 3100] and [600, 3500], respectively. We cropped the training images into 192×192 patches to train all the models. Data augmentation was performed by randomly rotating $90°$, $180°$, $270°$ and horizontally flipping the input. The mini-batch size in all the experiments was set to 16.

All SISR models were initialized using the method in [19]. The Adam solver [26] with the default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$) was adopted to optimize the network parameters. The learning rate was fixed at 10^{-4} and all the networks were trained for $1,000K$ iterations. All the comparing models were trained using the Caffe [22] toolbox, and tested using Caffe MATLAB interface. All the experiments were conducted on a PC equipped with an Intel Core i7-7820X CPU, 128G RAM and a single Nvidia Quadro GV100 GPU (32G). Our dataset and source code can be downloaded at https://github.com/csjcai/RealSR.

5.1. Simulated SISR datasets vs. RealSR dataset

To demonstrate the advantages of our RealSR dataset, we conduct experiments to compare the real-world super-resolution performance of SISR models trained on simulated datasets and RealSR dataset. Considering that most state-of-the-art SISR models were trained on DIV2K [46] dataset, we employed the DIV2K to generate simulated image pairs with bicubic degradation (BD) and multiple degradations (MD) [62]. We selected three representative and state-of-the-art SISR networks, *i.e.*, VDSR [24], SRResNet [30] and RCAN [64], and trained them on the BD, MD and RealSR training datasets for each of the three scaling factors ($\times 2$, $\times 3$, $\times 4$), leading to a total of 27 SISR models. To keep the network structures of SRResNet and RCAN unchanged, the input images were shuffled with factor $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$ for the three scaling factors $\times 2$, $\times 3$, $\times 4$, respectively.

We applied the 27 trained SISR models to the RealSR testing set, and the average PSNR and SSIM indices are listed in Table 2. The baseline bicubic interpolator is also included for comparison. One can see that, on our RealSR testing set, the VDSR and SRResNet models trained on the simulated BD dataset can only achieve comparable performance to the simple bicubic interpolator. Training on the MD dataset brings marginal improvements over BD, which indicates that the authentic degradation in real-world images is difficult to simulate. Employing a deeper architecture, the RCAN (>400 layers) can improve (0.2dB∼0.3dB) the performance over VDSR and SRResNet on all cases.

Using the same network architecture, SISR models trained on our RealSR dataset obtain significantly better performance than those trained on BD and MD datasets for all the three scaling factors. Specifically, for scaling factor $\times 2$, the models trained on our RealSR dataset have about 1.0dB improvement on average for all the three network architectures. The advantage is also significant for scaling factors $\times 3$ and $\times 4$. In Fig. 5, we visualize the super-resolved images obtained by different models. As can be seen, the SISR results generated by models trained on simulated BD and MD datasets tend to have blurring edges with obvious artifacts. On the contrary, models trained on our RealSR dataset recover clearer and more natural image details. More visual examples can be found in the **supplementary file**.

5.2. SISR models trained on RealSR dataset

To demonstrate the efficiency and effectiveness of the proposed LP-KPN, we then compare it with 8 SISR mod-

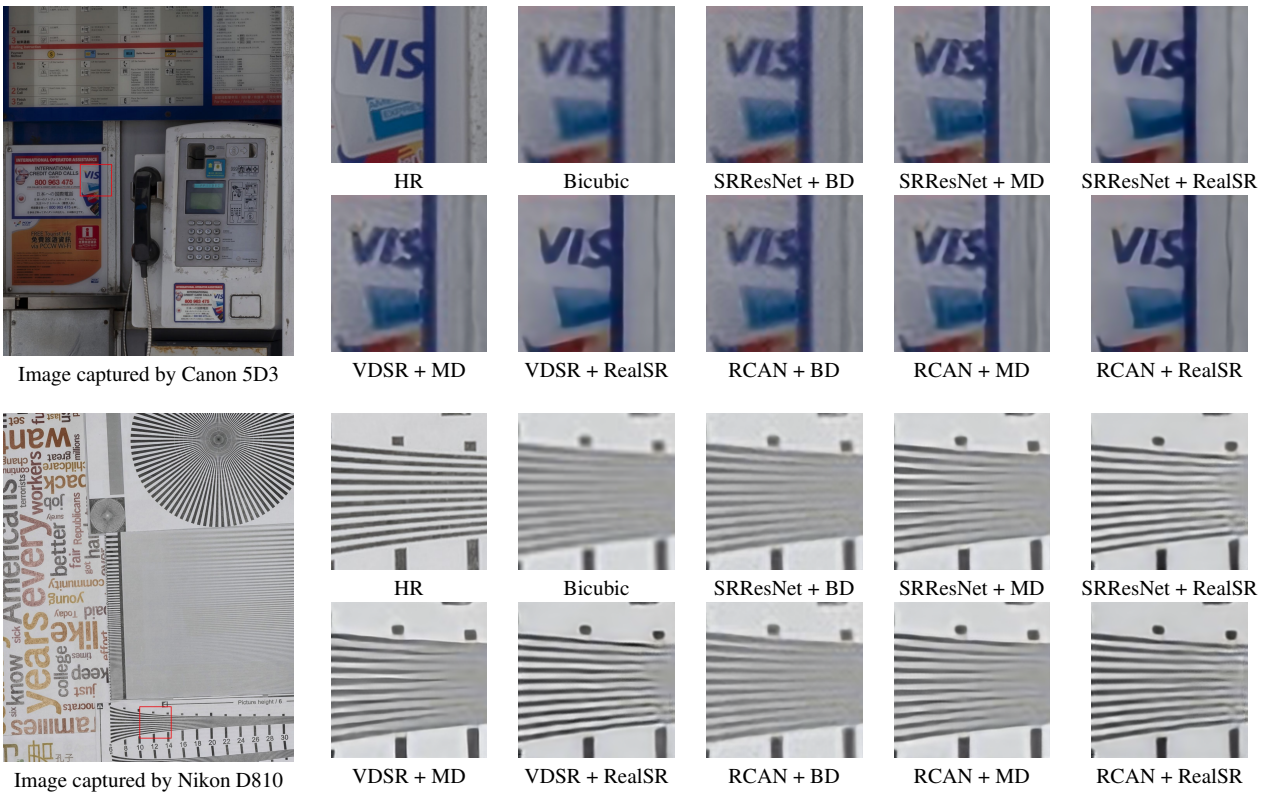


Figure 5. SR results ($\times 4$) on our RealSR testing set by different methods (trained on different datasets).

Table 3. Average PSNR (dB) and SSIM indices for different models (trained on our RealSR training set) on our RealSR testing set.

Method	PSNR			SSIM		
	$\times 2$	$\times 3$	$\times 4$	$\times 2$	$\times 3$	$\times 4$
Bicubic	32.61	29.34	27.99	0.907	0.841	0.806
VDSR	33.64	30.14	28.63	0.917	0.856	0.821
SRResNet	33.69	30.18	28.67	0.919	0.859	0.824
RCAN	33.87	30.40	28.88	0.922	0.862	0.826
DPS	33.71	30.20	28.69	0.919	0.859	0.824
KPN, $k = 5$	33.75	30.26	28.74	0.920	0.860	0.826
KPN, $k = 7$	33.78	30.29	28.78	0.921	0.861	0.827
KPN, $k = 13$	33.83	30.35	28.85	0.923	0.862	0.828
KPN, $k = 19$	33.86	30.39	28.90	0.924	0.864	0.830
Our, $k = 5$	33.90	30.42	28.92	0.927	0.868	0.834

els, including VDSR, SRResNet, RCAN, a baseline direct pixel synthesis (DPS) network and four KPN models with kernel size $k = 5, 7, 13, 19$. The DPS and the four KPN models share the same backbone as our LP-KPN. All models are trained and tested on our RealSR dataset. The PSNR and SSIM indices of all the competing models as well as the bicubic baseline are listed in Table 3.

One can notice that among the four direct pixel synthesis networks (*i.e.*, VDSR, SRResNet, RCAN and DPS), RCAN obtains the best performance because of its very deep architecture (over 400 layers). Using the same backbone with less than 50 layers, the KPN with 5×5 kernel size already outperforms the DPS. Using larger kernel size consistently brings better results for the KPN architecture, and it obtains comparable performance to the RCAN when the kernel size increases to 19. Benefitting from the Laplacian pyramid decomposition strategy, our LP-KPN using three different 5×5 kernels achieves even better results than the KPN with 19×19 kernel. The proposed LP-KPN obtains the best performance but with the lowest computational cost for all the three scaling factors. The detailed complexity analysis and visual examples of the SISR results by the competing models can be found in the **supplementary file**.

5.3. Cross-camera testing

To evaluate the generalization capability of SISR models trained on our RealSR dataset, we conduct a cross-camera testing. Images taken by two cameras are divided into training and testing sets, separately, with 15 testing images for each camera at each scaling factor. The three scales of images are combined for training, and models trained on one camera are tested on the testing sets of both cameras. The LP-KPN and RCAN models are compared in this evaluation, and the PSNR indexes are reported in Table 4.

It can be seen that for both RCAN and LP-KPN, the cross-camera testing results are comparable to the in-camera setting with only about 0.32dB and 0.30dB gap, respectively, while both are much better than bicubic interpolator. This indicates that the SISR models trained on one camera can generalize well to the other camera. This is possibly because our RealSR dataset contains various degradations produced by the camera lens and image formation

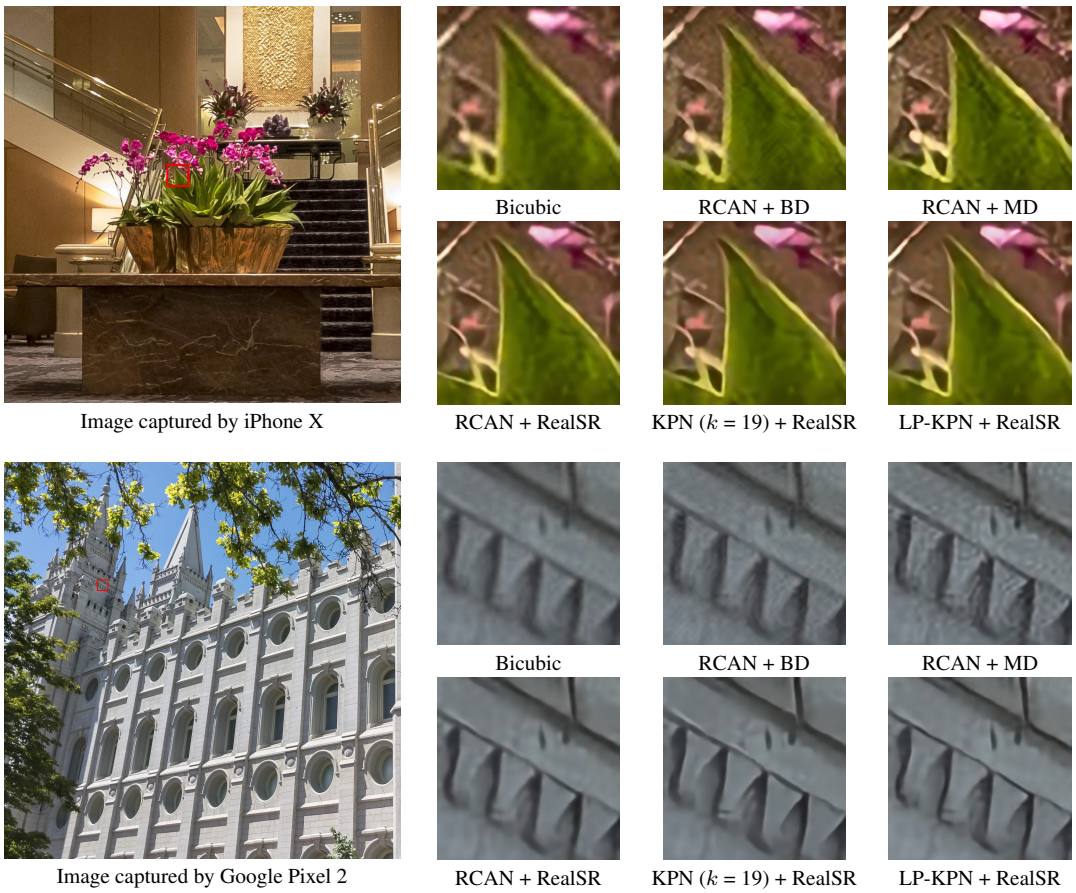


Figure 6. SISR results ($\times 4$) of real-world images outside our dataset. Images are captured by iPhone X and Google Pixel 2.

Table 4. Average PSNR (dB) index for cross-camera evaluation.

Tested	Scale	Bicubic	RCAN (Trained)		LP-KPN (Trained)	
			Canon	Nikon	Canon	Nikon
Canon	$\times 2$	33.05	34.34	34.11	34.38	34.18
	$\times 3$	29.67	30.65	30.28	30.69	30.33
	$\times 4$	28.31	29.46	29.04	29.48	29.10
Nikon	$\times 2$	31.66	32.01	32.30	32.05	32.33
	$\times 3$	28.63	29.30	29.75	29.34	29.78
	$\times 4$	27.28	27.98	28.12	28.01	28.13

process, which share similar properties across cameras. Between RCAN and LP-KPN models, the former has more parameters and thus is easier to overfit to the training set, delivering slightly worse generalization capability than LP-KPN. Similar observation has been found in [1, 49, 35].

5.4. Tests on images outside our dataset

To further validate the generalization capability of our RealSR dataset and LP-KPN model, we evaluate our trained model as well as several competitors 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). Since there are no ground-truth HR versions of these images, we visualize the super-resolved results in Fig. 1 and Fig. 6. In all these cases, the LP-KPN trained on our RealSR dataset obtains better visual quality than the competitors, recovering more natural and clearer details. More examples can be found in the **supplementary file**.

6. Conclusion

It has been a long standing problem for SISR research that the models trained on simulated datasets can hardly generalize to real-world images. We made an attempt to address this issue, and constructed a real-world super-solution (RealSR) dataset with authentic degradations. One Canon and one Nikon cameras were used to collect 595 HR and LR image pairs, and an effective image registration algorithm was developed to ensure accurate pixel-wise alignment between image pairs. A Laplacian pyramid based kernel prediction network was also proposed to perform efficient and effective real-world SISR. Our extensive experiments validated that the models trained on our RealSR dataset can lead to much better real-world SISR results than trained on existing simulated datasets, showing good generalization capability to other cameras. In the future, we will enlarge the RealSR dataset by collecting more image pairs with more types of cameras, and investigate new SISR model training strategies on it.

References

[1] S. Bako, T. Vogels, B. McWilliams, M. Meyer, J. Novák, A. Harvill, P. Sen, T. Derose, and F. Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Trans. Graph.*, 36(4):97–1, 2017. 2, 3, 5, 8

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006. 4

[3] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 2

[4] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor. The 2018 PIRM challenge on perceptual image super-resolution. In *ECCV*, 2018. 1

[5] A. Bulat, J. Yang, and G. Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first. In *ECCV*, 2018. 2

[6] J. Cai, S. Gu, R. Timofte, and L. Zhang. Ntire 2019 challenge on real image super-resolution: Methods and results. In *CVPRW*, 2019. 1

[7] C. Chang, X. Zhiwei, T. Xinmei, Z. Zheng-Jun, and W. Feng. Camera lens super-resolution. In *CVPR*, 2019. 3

[8] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004. 1

[9] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *ICASSP*, 2008. 4

[10] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1

[11] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. 2

[12] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011. 1

[13] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution. In *ICCV*, 2013. 2

[14] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG)*, 30(2):12, 2011. 1

[15] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International journal of computer vision*, 40(1):25–47, 2000. 1

[16] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, pages 349–356, 2009. 1

[17] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *CVPR*, 2017. 2, 3, 5

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPs*, 2014. 2

[19] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[20] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*. 1, 2

[21] K. Jia, X. Wang, and X. Tang. Image transformation based on learning dictionaries across image spaces. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):367–380, 2013. 1

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014. 6

[23] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1

[24] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1, 6

[25] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1127–1133, 2010. 1

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[27] T. Köhler, M. Bätz, F. Naderi, A. Kaup, A. Maier, and C. Riess. Bridging the simulated-to-real gap: Benchmarking super-resolution on real data. *arXiv preprint arXiv:1809.06420*, 2018. 2, 3

[28] S. Kong and C. Fowlkes. Image reconstruction with predictive filter flow. *arXiv preprint arXiv:1811.11482*, 2018. 5

[29] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1

[30] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1, 6

[31] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017. 1, 5, 6

[32] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 4

[33] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 2

[34] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 2

[35] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll. Burst denoising with kernel prediction networks. In *CVPR*, 2018. 2, 3, 5, 8

[36] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017. 2, 3

[37] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017. 2, 3

[38] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of visual communication and image representation*, 6(4):348–365, 1995. 4

[39] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003. 1

[40] C. Qu, D. Luo, E. Monari, T. Schuchert, and J. Beyerer. Capturing ground truth super-resolution data. In *ICIP*, 2016. 2, 3

[41] M. S. Sajjadi, B. Scholkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, 2017. 1

[42] S. Schulter, C. Leistner, and H. Bischof. Fast and accurate image upscaling with super-resolution forests. In *CVPR*, 2015. 1

[43] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 1, 5

[44] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, 2015. 2, 3, 5

[45] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *CVPR*, 2017. 1

[46] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, 2017. 1, 2, 6

[47] R. Timofte, V. De Smet, and L. Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*, 2013. 1

[48] R. Timofte, S. Gu, J. Wu, and L. Van Gool. Ntire 2018 challenge on single image super-resolution: methods and results. In *CVPRW*, 2018. 1, 5, 6

[49] T. Vogels, F. Rousselle, B. McWilliams, G. Röthlin, A. Harvill, D. Adler, M. Meyer, and J. Novák. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)*, 37(4):124, 2018. 2, 3, 5, 8

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[51] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *ICCV*, 2015. 1

[52] Wikipedia contributors. Circle of confusion — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Circle_of_confusion&oldid=885361501, 2019. [Online; accessed 6-March-2019]. 5

[53] Wikipedia contributors. Depth of field — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Depth_of_field&oldid=886691980, 2019. [Online; accessed 6-March-2019]. 3, 5

[54] Wikipedia contributors. Lens (optics) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Lens_(optics)&oldid=882234499, 2019. [Online; accessed 6-March-2019]. 3

[55] R. G. Willson and S. A. Shafer. What is the center of the image? *JOSA A*, 11(11):2946–2955, 1994. 4

[56] C.-Y. Yang, C. Ma, and M.-H. Yang. Single-image super-resolution: A benchmark. In *ECCV*, 2014. 2

[57] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. In *CVPR*. 1

[58] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. 1

[59] W. Yang, X. Zhang, Y. Tian, W. Wang, and J.-H. Xue. Deep learning for single image super-resolution: A brief review. *arXiv preprint arXiv:1808.03344*, 2018. 1

[60] H. Yong, D. Meng, W. Zuo, and L. Zhang. Robust online matrix factorization for dynamic background subtraction. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1726–1740, 2018. 4

[61] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 2

[62] K. Zhang, W. Zuo, and L. Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, 2018. 1, 2, 6

[63] X. Zhang, Q. Chen, R. Ng, and V. Koltun. Zoom to learn, learn to zoom. In *CVPR*, 2019. 3

[64] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1, 2, 6

[65] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1, 6