Blind Super-Resolution With Iterative Kernel Correction

Jinjin Gu[1]*, Hannan Lu[2]*, Wangmeng Zuo[2], Chao Dong[3]
[1]The School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[3]ShenZhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
jinjingu@link.cuhk.edu.cn, {hannanlu, wmzuo}@hit.edu.cn, chao.dong@siat.ac.cn

Abstract

Deep learning based methods have dominated super-resolution (SR) field due to their remarkable performance in terms of effectiveness and efficiency. Most of these methods assume that the blur kernel during downsampling is predefined/known (e.g., bicubic). However, the blur kernels involved in real applications are complicated and unknown, resulting in severe performance drop for the advanced SR methods. In this paper, we propose an Iterative Kernel Correction (IKC) method for blur kernel estimation in blind SR problem, where the blur kernels are unknown. We draw the observation that kernel mismatch could bring regular artifacts (either over-sharpening or over-smoothing), which can be applied to correct inaccurate blur kernels. Thus we introduce an iterative correction scheme – IKC that achieves better results than direct kernel estimation. We further propose an effective SR network architecture using spatial feature transform (SFT) layers to handle multiple blur kernels, named SFTMD. Extensive experiments on synthetic and real-world images show that the proposed IKC method with SFTMD can provide visually favorable SR results and the state-of-the-art performance in blind SR problem.

1. Introduction

As a fundamental low-level vision problem, single image super-resolution (SISR) is an active research topic and has attracted increasingly attention. SISR aims to reconstruct the high-resolution (HR) image from its low-resolution (LR) observation. Since the seminal work of employing convolutional neural networks (CNNs) for SR [6], various deep learning based methods with different network architectures [15, 16, 18, 29, 41, 10, 40] and training strategies [19, 34, 27, 5] have been proposed to continuously improve the SR performance. Most of the existing advanced SR methods assume that the downsampling blur kernel is known and pre-defined, but the blur kernels involved in real applications are typically complicated and unavailable. As has been revealed in [9, 36], learning-based methods will suffer severe performance drop when the pre-defined blur kernel is different from the real one. This phenomenon of kernel mismatch will introduce undesired artifacts to output images, as shown in Figure 2. Thus the problem with unknown blur kernels, also known as *blind SR*, has failed most of deep learning based SR methods and largely limited their usage in real-world applications.

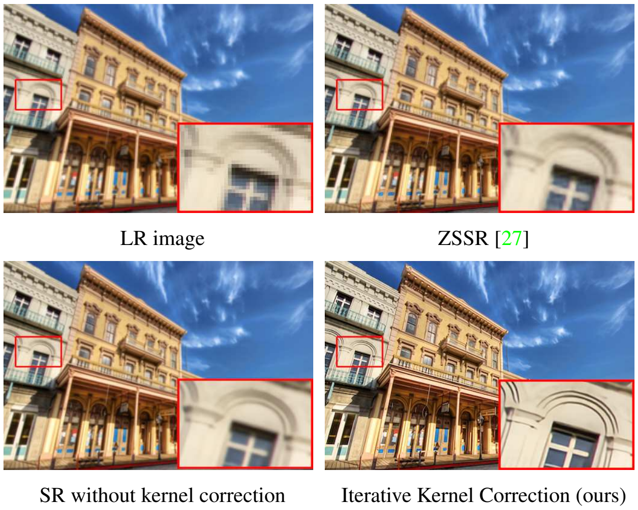

LR image
ZSSR [27]
SR without kernel correction
Iterative Kernel Correction (ours)

Figure 1. SISR results on image "*img_017*" with SR factor 4. Before bicubic downsamping, the HR image is blurred by a Gaussian kernel with $\sigma = 1.8$

Most existing blind SR methods are model-based [3, 32, 11, 12, 14], which usually involve complicated optimization procedures. They predict the underlying blur kernel using self-similarity properties of natural images [23]. However, their predictions are easily affected by input noises, leading to inaccurate kernel estimation. A few deep learning based methods have also tried to make progress for blind SR. For example, in CAB [25] and SRMD [39], the net-

*This work was done when they were interns at SenseTime.

work can take the blur kernel as an additional input and generate different results according to the provided kernel. They achieve satisfactory performance if the input kernel is close to the ground truth. However, these methods still cannot predict the blur kernel for every image on hand, thus are not applicable in real applications. Although deep learning based methods have dominated SISR, they have limited progress on blind SR problem.

In this paper, we focus on using deep learning methods to solve the blind SR problem. Our method stems from the observation that artifacts caused by kernel mismatch have regular patterns. Specifically, if the input kernel is smoother than the real one, then the output image will be blurry/over-smoothing. Conversely, if the input kernel is sharper than the correct one, then the results will be over-shapened with obvious ringing effects (see Figure 2). This asymmetry of kernel mismatch effect provides us an empirical guidance on how to correct an inaccurate blur kernel. In practical, we propose an Iterative Kernel Correction (IKC) method for blind SR based on predict-and-correct principle. The estimated kernel is iteratively corrected by observing the previous SR results, and gradually approaches the ground truth. Even the predicted blur kernel is slightly different from the real one, the output image can still get rid of those regular artifacts caused by kernel mismatch.

By further diving into the SR methods proposed for multiple blur kernels (i.e., SRMD [39]), we find that taking the concatenation of image and blur kernel as input is not the optimal choice. To make a step forward, we employ spatial feature transform (SFT) layers [33] and propose an advanced CNN structure for multiple blur kernels, namely SFTMD. Experiments demonstrate that the proposed SFTMD is superior to SRMD by a large margin. By combining the above components – SFTMD and IKC, we achieve state-of-the-art (SOTA) performance on blind SR problem.

We summarize our contributions as follows: (1) We propose an intuitive and effective deep learning framework for blur kernel estimation in single image super resolution. (2) We propose a new non-blind SR network using the spatial feature transform layers for multiple blur kernels. We demonstrate the superior performance of the proposed non-blind SR network. (3) We test the blind SR performance on both carefully selected blur kernels and real images. Extensive experiments show that the combination of SFTMD and IKC achieves the SOTA performance in blind SR problem.

2. Related Work

Super-Resolution Neural Networks. In the past few years, neural networks have shown remarkable capability on improving SISR performance. Since the pioneer work of using CNN to learn the end-to-end mapping from LR to HR images [6], plenty of CNN architectures have been proposed for SISR [7, 26, 18, 10, 16, 28]. In order to go deeper in network depth and achieve better performance, most of the existing high-performance SR networks have residual architecture [15]. SRGAN [19] first introduce residual blocks into SR networks. EDSR [20] improve it by removing unnecessary batch normalization layer in residual block and expanding the model size. DenseSR [41] present an effective residual dense block and ESRGAN [34] further use a residual-in-residual dense block to improve the perceptual quality of SR results. Zhang *et al.* [40] introduce the channel attention component in residual blocks. Some networks are specifically designed for the SR task in some special scenarios, e.g., Wang *et al.* [33] use a novel spatial feature transform layer to introduce the semantic prior as an additional input of SR network. Moreover, Riegler *et al.* [25] propose conditioned regression models can effectively exploit the additional kernel information during training and inference. SRMD [39] propose a stretching strategy to integrate non-image degradation information in a SR network.

Blind Super-Resolution. Blind SR assume that the degradation kernels are unavailable. In recent years, the community has paid relatively less research attention to blind SR problem. Michaeli and Irani [23] estimate the optimal blur kernel based on the property that small image patches will re-appear in images. There are also research works trying to employ deep learning in blind SR task. Yuan *et al.* [37] propose to learn not only SR mapping but also the degradation mapping using unsupervised learning. Shocher *et al.* [27] exploit the internal recurrence of information inside an image and propose an unsupervised SR method to super-resolve images with different blur kernels. They train a small CNN on examples extracted from the input image itself, the trained image-specific CNN is appropriate for super-resolving this image. Different from the previous works, our method employs the correlation between SR results and kernel mismatch. Our method uses the intermediate SR results to iteratively correct the estimation of blur kernels, thus provide artifact-free final SR results.

3. Method

3.1. Problem Formulation

The blind super-resolution problem is formulated as follows. Mathematically, the HR image I^{HR} and LR image I^{LR} are related by a degradation model

$$I^{LR} = (k \otimes I^{HR}) \downarrow_s + n, \tag{1}$$

where $\otimes$ denotes convolution operation. There are three main components in this model, namely the blur kernel k, the downsampling operation $\downarrow_s$ and the additive noise n. In literature, the most widely adopted blur kernel is isotropic Gaussian blur kernel [8, 36, 39]. Besides, the anisotropic blur kernels also appear in some works [25, 39], which can be regarded as the combination of a motion blur and an isotropic blur kernel. For simplicity, we mainly focus on the isotropic blur kernel without motion effect in this paper. Following most recent deep learning based SR methods [39], we adopt the combination of Gaussian blur and bicubic downsampling. In real-world use cases, the LR images are often accompanied with additive noises. As in SRMD [39], we assume that the additive noise follows Gaussian distribution in real world application. Note that the formulation of blind SR in this paper is different with the previous works [23, 37] . Although defined as blind SR problem, our method focuses on a limited variety of kernels and noise. But the kernel estimated according to our assumptions can handle most of the real world images.

3.2. Motivation

We then review the importance of using correct blur kernel during SISR based on the settings described above. In order to obtain the LR images I^{LR}, the HR images I^{HR} are first blurred by the isotropic Gaussian kernel with kernel width σ_{LR} and then downsampled by bicubic interpolation. Assume that the mapping $\mathcal{F}(I^{LR}, k)$ is a well-trained SR model with the kernel information as input (e.g., SRMD [39]). Then the output image is artifact-free with correct kernel k. The blind SR problem is equivalent to finding the kernel k that helps SR model generate visual pleasing result I^{SR}. A straightforward solution is to adopt a predictor function $k' = \mathcal{P}(I^{LR})$ that estimates k from the LR input directly. The predictor can be optimized by minimizing the l_2 distance as

$$\theta_{\mathcal{P}} = \arg\min_{\theta_{\mathcal{P}}} \|k - \mathcal{P}(I^{LR}; \theta_{\mathcal{P}})\|_2^2, \tag{2}$$

where $\theta_{\mathcal{P}}$ is the parameter of $\mathcal{P}$. By employing the predictor function and the SR model together, we are able to build an end-to-end blind SR model.

However, accurate estimation of k is impossible. As the inverse problem is ill-posed, there exists multiple candidates of k for a single input. Meanwhile, the SR models are very sensitive to the estimation error. If the inaccurate kernel is used for SR directly, then the final SR results will contain obvious artifacts. Figure 2 shows the sensitivity of the SR results to kernel mismatch, where σ_{SR} denotes the kernel width used for SR. As shown in the upper-right region of Figure 2, where the kernel used for SR are sharper than the real one ($\sigma_{SR} < \sigma_{LR}$), the SR results are over-smoothing and the the high frequency textures are significantly blurred. In the lower-left region of Figure 2, where the kernel used for SR are smoother than the correct one ($\sigma_{SR} > \sigma_{LR}$), the SR results show unnatural ringing artifacts caused by over-enhancing high-frequency edges. In contrast, the results on the diagonal, which use correct blur kernels, look natural without artifacts and blurring. The above phenomenon illustrates that the estimation error of k will be significantly magnified by the SR model, resulting in unnatural output images. To address the kernel mismatch problem, we propose to iteratively correct the kernel until we obtain an artifact-free SR results.

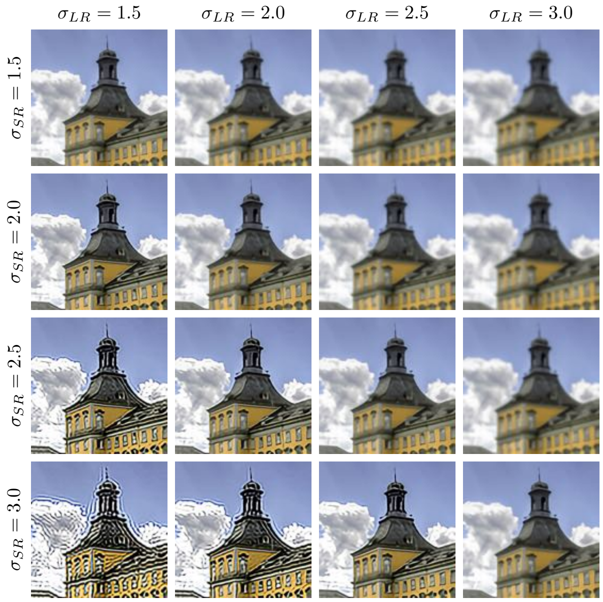


Figure 2. SR sensitivity to the kernel mismatch. Where σ_{LR} denotes the kernel used for downsampling and σ_{SR} denotes the kernel used for SR.

To correctly estimate k, we build a corrector function $\mathcal{C}$ that measures the difference between the estimated kernel and the ground truth kernel. In the core of our idea is to adopt the intermediate SR results. The corrector function can be obtained by minimizing the l_2 distance between the corrected kernel and the ground truth as

$$\theta_{\mathcal{C}} = \arg\min_{\theta_{\mathcal{C}}} \|k - (\mathcal{C}(I^{SR}; \theta_{\mathcal{C}}) + k')\|_2^2, \tag{3}$$

where $\theta_{\mathcal{C}}$ is the parameter of $\mathcal{C}$ and I^{SR} is the SR result using the last estimated kernel. This corrector adjusts the estimated blur kernel based on the features of the SR image. After correction, the SR results using adjusted kernel are supposed to approach natural images with less artifacts.

However, if we train our model with only one time of correction, the corrector may provide inadequate correction or over-correct the kernel, leading to unsatisfactory SR results. A possible solution is to use smaller correction steps that gradually refine the kernel until it reaches ground truth. When the SR result does not contain serious over-smoothing or over-sharpening effects, the corrector will make little changes to the estimated kernel to ensure convergence. Then we are able to get a high-quality SR image by iteratively applying kernel correction. Experiments also demonstrate our assumption. Figure 3 shows the PSNR and SSIM results using different iteration numbers.

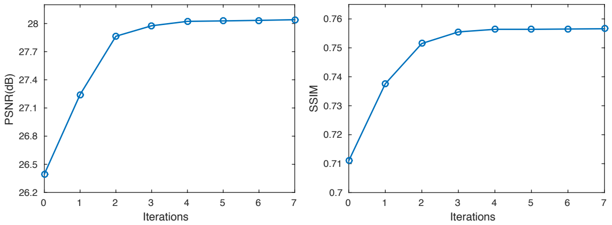


Figure 3. The curves of PSNR and SSIM vs. iterations. The experiments are conducted using IKC method. The test set is Set14 and the SR factor is 4.

It can be observed that correcting only once is not sufficient. When the number of iterations increases, both PSNR and SSIM increase gradually until convergence.

3.3. Proposed Method

Overall framework. The proposed Iterative Kernel Correction (IKC) framework consists of a SR model $\mathcal{F}$, a predictor $\mathcal{P}$ and a corrector $\mathcal{C}$, and the pseudo-code is shown in Algorithm 1. Suppose the LR image I^{LR} is of size $C \times H \times W$, where C denotes the number of channels, H and W denote the height and width of the image. We assume that blur kernel is of size $l \times l$ and the kernel space is a l^2-dimensional linear space. In order to save computation, we first reduce the dimensionality of the kernel space by principal component analysis (PCA). The kernels are projected onto a b-dimensional linear space by a dimension reduction matrix $M \in \mathbb{R}^{b \times l^2}$. Thus we only need to perform estimation in this low dimensional space, which is more effective in calculation. The kernel after the dimension reduction is denoted by h, where $h = Mk, h \in \mathbb{R}^b$. At the start of the algorithm, an initial estimation h_0 is given by the predictor function $h_0 = \mathcal{P}(I^{LR})$, and then used to get the first SR result $I_0^{SR} = \mathcal{F}(I^{LR}, h_0)$. After obtaining the initial estimation, we proceed to the correction phase of the estimated kernel. At the ith iteration, given the previous estimation h_{i-1}, the correcting update Δh_i, the new estimation h_i and the new SR result I_i^{SR} can be written as

$$\Delta h_i = \mathcal{C}(I_i^{SR}, h_{i-1}) \tag{4}$$

$$h_i = h_{i-1} + \Delta h_i \tag{5}$$

$$I_i^{SR} = \mathcal{F}(I^{LR}, h_i). \tag{6}$$

After t iterations, the I_t^{SR} is the final output of IKC.

Network architecture of SR model $\mathcal{F}$. As the most successful SR method for multiple blur kernels, SRMD [39] propose a simple yet efficient stretching strategy for CNN to process non-image input directly. SRMD stretches the input h into kernel maps $\mathcal{H}$ of size $b \times H \times W$, where all the elements of the ith map are equal to the ith element of h. SRMD takes the concatenated LR image and kernel maps of size $(b+C) \times H \times W$ as input. Then, a cascade of 3×3 convolution layers and one pixel-shuffle upsampling layer are applied to perform super-resolution. However, to exploit the kernel information, concatenating the image and the transformed kernel as input is not the only or best choice. On the one hand, the kernel maps do not actually contain the information of the image. Processing the kernel maps and the image at the same time with convolution operation will introduce interference that is not related to the image. Using this concatenation strategy with residual blocks can interfere with image processing, making it difficult to employ residual structure to improve performance. On the other hand, the influence of kernel information is only considered at the first layer. When applying the same strategy in a deeper network, the deeper layers are difficult to be affected by the kernel information input at the first layer. To address above problems, we proposed a new SR model for multiple kernels using spatial feature transform (SFT) layers [33], namely SFTMD. In SFTMD, the kernel maps influence the output of network by applying an affine transformation to the feature maps in each middle layer by SFT layers. This affine transformation is not involved in the process of input image directly, thus providing better performance.

Algorithm 1 Iterative Kernel Correction

```
Require: the LR image $I^{LR}$
Require: the max iteration number $t$
$h_0 \leftarrow \mathcal{P}(I^{LR})$ (Initialize the kernel estimation)
$I_0^{SR} \leftarrow \mathcal{F}(I^{LR}, h_0)$ (The initial SR result)
$i \leftarrow 0$ (Initialize counter)
while $i < t$ do
    $i \leftarrow i + 1$
    $\Delta h_i \leftarrow \mathcal{C}(I_{i-1}^{SR}, h_{i-1})$ (Estimate the kernel error using the intermediate SR results)
    $h_i \leftarrow h_{i-1} + \Delta h_i$ (Update kernel estimation)
    $I_i^{SR} \leftarrow \mathcal{F}(I^{LR}, h_i)$ (Update the SR result)
return $I_t^{SR}$ (Output the final SR result)
```

Figure 4 illustrates the network architecture of SFTMD. We employ the high level architecture of SRResNet [19] and extend it to handle multiple kernels by SFT layers. The SFT layer provides affine transformation for the feature maps F conditioned on the kernel maps $\mathcal{H}$ by a scaling and shifting operation:

$$\mathrm{SFT}(F, \mathcal{H}) = \gamma \odot F + \beta, \tag{7}$$

where γ and β is the parameters for scaling and shifting, $\odot$ present Hadamard product. The transformation parameters γ and β are obtained by small CNN. Suppose that the output feature maps of the previous layer F are of size $C_f \times H \times W$, where C_f is the number of feature maps, and the kernel maps are of size $b \times H \times W$. The CNN takes the concatenated feature maps and kernel maps (total size is $(b + C_f) \times H \times W$) as input and output γ and β. We use SFT layers after all convolution layers in residual blocks

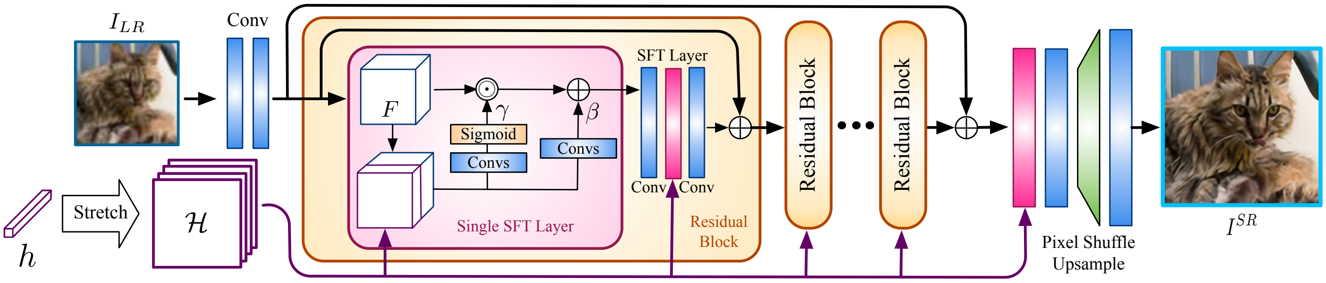


Figure 4. The architecture of the proposed SFTMD network. The design of the proposed SFT layer is shown in pink box.

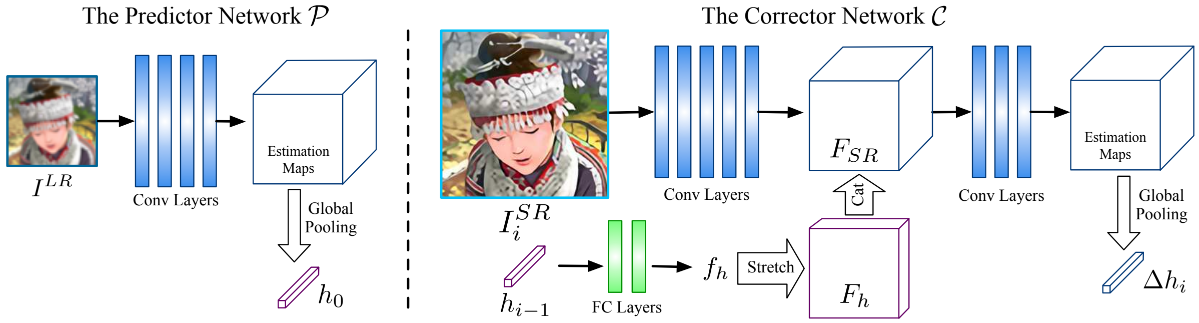


Figure 5. The network architecture of the proposed predictor and corrector.

and after the global residual connection. It is worth pointing out that the code maps are spatially uniform, thus the SFT layers do not actually provide spatial variability according to the code maps. This is different from its application in semantic super resolution [33]. We only employ the transformation characteristic of SFT layers.

Network architecture of predictor $\mathcal{P}$ and corrector $\mathcal{C}$. The network designs of the predictor and corrector are shown in Figure 5. For the predictor $\mathcal{P}$, we use four convolution layers with Leaky ReLU activations and a global average pooling layer. The convolution layers give the estimation of the kernel h spatially and form the estimation maps. Then the global average pooling layer gives the global estimation by taking the mean value spatially.

For the corrector $\mathcal{C}$, we take not only the SR image I^{SR} but also the previous estimation h as inputs. Similar to Eq. (3), the new corrector can be obtained by solving the following optimization problem:

$$\theta_{\mathcal{C}} = \arg\min_{\theta_{\mathcal{C}}} \|k - (\mathcal{C}(I^{SR}, h; \theta_{\mathcal{C}}) + k')\|_2^2. \qquad (8)$$

The input SR result is first processed to feature maps F_{SR} by five convolution layers with Leaky ReLU activations. Note that the previous SR result may contain artifacts (e.g., ringing and blurry) caused by kernel mismatch, which can be extracted by these convolution layers. At the same time, we use two fully-connected layers with Leaky ReLU activations to extract the inner correlations of the previous kernel estimation. We then stretch the output vector f_h to feature maps F_h using the same strategy used in SFTMD. The F_h and F_{SR} are then concatenated to predict the Δh. We use three convolution layers with kernel size 1×1 and Leaky ReLU activations to give the estimation for Δh spatially. Same as the predictor, a global average pooling operation is used to get the global estimation of Δh.

4. Experiments

4.1. Data Preparation and Network Training

We synthesize the training image pairs according to the problem formulation described in section 3.1. For the isotropic Gaussian blur kernels used for training, the kernel width ranges are set to $[0.2, 2.0]$, $[0.2, 3.0]$ and $[0.2, 4.0]$ for SR factors 2, 3 and 4, respectively. We uniformly sample the kernel width in the above ranges. The kernel size is fixed to 21×21. When applying on real world images, we use the additive Gaussian noise with covariance $\sigma = 15$. We also provide noise-free version for comparison on the synthetic test images. The HR images are collected from DIV2K [1] and Flickr2K [30], then the training set consists of 3450 high-quality 2K images. The training dataset is augmented with random horizontal flips and 90 degree rotations. All models are trained and tested on RGB channels.

The SFTMD and IKC are both trained on the synthetic training image pairs and their corresponding blur kernels. First, the SFTMD is pre-trained using mean square error (MSE) loss. We then train the predictor network and the corrector network alternately. The parameters of the trained SFTMD are fixed during training the predictor and the corrector. The order of training can refer to Algorithm 1. For every mini-batch data $\{I_i^{LR}, I_i^{HR}, h_i\}_{i=1}^N$, where N denotes the mini-batch size, we first update the parameters of the predictor according to Eq. (2). We then update the cor-

Table 1. Quantitative comparison of SRCNN-CAB [25], SRMDNF [39] and the proposed SFTMD. The comparison is conducted using three different isotropic Gaussian kernels on Set5, Set14 and BSD100 dataset. The best two results are highlighted in red and blue colors.

Method	Kernel Width	Set5 [4]			Set14 [38]			BSD100 [21]		
		×2	×3	×4	×2	×3	×4	×2	×3	×4
SRCNN-CAB [25]	0.2	33.27	31.03	29.31	30.29	28.29	26.91	28.98	27.65	25.51
SRMDNF [39]		37.79	34.13	31.96	33.33	30.04	28.35	32.05	28.97	27.49
SRResNet, concatenate at the first layer		31.74	30.90	29.40	27.57	26.40	26.18	27.24	26.43	26.34
SRResNet, replace SFT layer by direct concatenation		37.69	34.01	31.64	33.26	30.04	28.23	31.83	28.81	27.26
SFTMD (ours)		38.00	34.57	32.39	33.68	30.47	28.77	32.09	29.09	27.58
SRCNN-CAB [25]	1.3	33.42	31.14	29.50	30.51	28.34	27.02	29.02	27.91	25.66
SRMDNF [39]		37.44	34.17	32.00	33.20	30.08	28.42	31.98	29.03	27.53
SRResNet, concatenate at the first layer		30.88	30.33	29.11	27.16	25.84	25.93	26.84	25.92	26.20
SRResNet, replace SFT layer by direct concatenation		37.01	34.02	31.69	32.96	30.13	28.29	31.58	28.89	27.29
SFTMD (ours)		38.00	34.57	32.39	33.68	30.47	28.77	32.09	29.09	27.58
SRCNN-CAB [25]	2.6	32.21	30.82	28.81	29.74	27.83	26.15	28.35	26.63	25.13
SRMDNF [39]		34.12	33.02	31.77	30.25	29.33	28.26	29.23	28.35	27.43
SRResNet, concatenate at the first layer		24.22	28.44	28.64	22.99	24.19	25.63	23.07	24.42	25.99
SRResNet, replace SFT layer by direct concatenation		27.75	32.71	31.35	25.67	29.28	28.07	25.57	28.19	27.15
SFTMD (ours)		38.00	34.57	32.39	33.68	30.47	28.77	32.09	29.09	27.58

rector according to Eq. (8) with a fixed iteration number $t = 7$. For optimization, we use Adam [17] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate 1×10^{-4}. We implement our models with the PyTorch framework and train them using NVIDIA Titan Xp GPUs.

We also propose a test kernel set for the quantitative evaluation of blind SR methods, namely *Gausssian8*. As declared by the name, *Gausssian8* consists eight selected isotropic Gaussian blur kernels for each SR factor 2, 3 and 4 (twenty four kernels in total). The ranges of kernel width are set to $[0.80, 1.60]$, $[1.35, 2.40]$ and $[1.80, 3.20]$ for SR factors 2, 3 and 4, respectively. The HR images are first blurred by the selected blur kernels and then downsampled by bicubic interpolation. By determining the blur kernels for testing, we can compare and analyze the performance of blind SR methods. Although it only contains isotropic Gaussian kernels, it can still be used to test the basic performance of a blind SR method.

4.2. Experiments of SFTMD

We evaluate the performance of the proposed SFTMD on different Gaussian kernels. The kernel settings are given in Table 1. We compare the SFTMD with the SOTA non-blind SR methods SRCNN-CAB [25] and SRMD [39]. As SFTMD adopts SRResNet as the main network, which is different from SRMD and SRCNN-CAB, we provide two additional baselines that have same network structures but different concatenation strategies: (1) SRResNet with concatenating $\mathcal{H}$ at the first layer, (2) SFTMD with SFT layer replaced by direct concatenation[1].Table 1 shows the quantitative comparison results. Comparing with the SOTA SR methods – SRCNN-CAB and SRMD, the proposed SFTMD achieves significantly better performance on all settings and dataset. Comparing with two additional baselines that all use SRResNet as the main network, SFTMD could also obtain the best results. This further demonstrated the effect of SFT layers. It is worth noting that directly concatenating $\mathcal{H}$ in SRResNet will cause severe performance drop. As the combination of direct concatenation strategy and residual structure will interfere with image processing and cause severe artifacts.

4.3. Experiments on Synthetic Test Images

We evaluate the performance of the proposed method on the synthetic test images. Figure 7 shows the intermediate results during correction. As one can see that the SR results using the kernel estimated by the predictor directly (the initial prediction in Figure 7) are unsatisfactory and contain either blurry or ringing artifacts. As the number of iterations increases, artifacts and blurring are gradually alleviated. The quantitative results (PSNR) also prove the necessity of the iterative correction strategy. We can see at the 4th iteration, the SR results using corrected kernels are able to show good visual quality.

We then conduct thorough comparisons with the SOTA non-blind and blind SR methods using *Gaussian8* kernels. We also provide the comparison with the solutions using the SOTA deblurring method. We perform blind debluring method Pan *et al.* [24] before and after the non-blind SR method CARN [2]. Table 2 shows the PSNR and SSIM [35] results on five widely-used datasets. As one can see, despite the remarkable performance under bicubic downsampling setting, the non-blind SR methods suffer severe performance drop when the downsampling kernel is different from the predefined bicubic kernel. The ZSSR [27] takes the effect of blur kernel into account, and provides better SR performance compared with non-blind SR methods. Performing blind deblurring on the LR images makes the SR images sharper, but lost in image quality The final SR results have severe distortion. Deblurring on the blurred super-resolved images provides better results, but fails to reconstruct textures and details. Although the SR results without kernel correction (denoted by "$\mathcal{P}$+SFTMD") achieves comparable quantitative performance with the existing methods, the SR performance can still be greatly im-

[1] Direct concatenation means concatenating the kernel maps with feature maps directly. This is different from the affine transformation in the SFT layer.

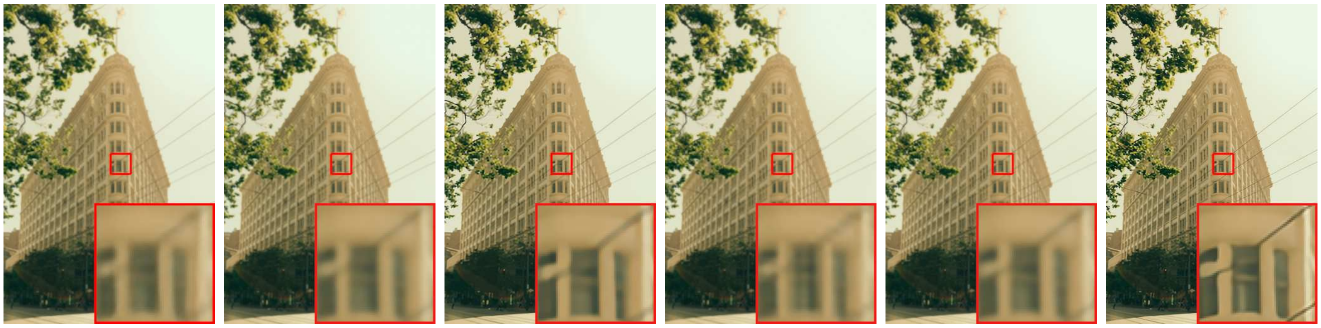

A+ [31] CARN [2] CARN + Pan *et al.*[24] ZSSR [27] $\mathcal{P}$+SFTMD IKC (ours)

Figure 6. SISR performance comparison of different methods with SR factor 4 and kernel width 1.8 on image "*Img_050*" from Urban100.

Table 2. Quantitative comparison of the SOTA SR methods and IKC method. The best two results are highlighted in red and blue colors, respectively. Note that the methods marked with "*" is not designed for blind SR, thus the comparison with these methods is unfair.

Method	Scale	Set5 [4]		Set14 [38]		BSD100 [21]		Urban100 [13]		Manga109 [22]	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	28.82	0.8577	26.02	0.7634	25.92	0.7310	23.14	0.7258	25.60	0.8498
CARN* [2]		30.99	0.8779	28.10	0.7879	26.78	0.7286	25.27	0.7630	26.86	0.8606
ZSSR [27]		31.08	0.8786	28.35	0.7933	27.92	0.7632	25.25	0.7618	28.05	0.8769
Pan *et al.* [24] + CARN [2]		24.20	0.7496	21.12	0.6170	22.69	0.6471	18.89	0.5895	21.54	0.7496
CARN [2] + Pan *et al.* [24]		31.27	0.8974	29.03	0.8267	28.72	0.8033	25.62	0.7981	29.58	0.9134
$\mathcal{P}$+ SFTMD		35.44	0.9617	31.27	0.8676	30.54	0.8946	27.80	0.8464	30.75	0.9074
IKC (ours)		36.62	0.9658	32.82	0.8999	31.36	0.9097	30.36	0.8949	36.06	0.9474
Bicubic	×3	26.21	0.7766	24.01	0.6662	24.25	0.6356	21.39	0.6203	22.98	0.7576
CARN* [2]		27.26	0.7855	25.06	0.6676	25.85	0.6566	22.67	0.6323	23.84	0.7620
ZSSR [27]		28.25	0.7989	26.11	0.6942	26.06	0.6633	23.26	0.6534	25.19	0.7914
Pan *et al.* [24] + CARN [2]		19.05	0.5226	17.61	0.4558	20.51	0.5331	16.72	0.4578	18.38	0.6118
CARN [2] + Pan *et al.* [24]		30.13	0.8562	27.57	0.7531	27.14	0.7152	24.45	0.7241	27.67	0.8592
$\mathcal{P}$+ SFTMD		31.26	0.9291	28.41	0.7811	27.37	0.8102	24.57	0.7458	26.29	0.8399
IKC (ours)		32.16	0.9420	29.46	0.8229	28.56	0.8493	25.94	0.8165	28.21	0.8739
Bicubic	×4	24.57	0.7108	22.79	0.6032	23.29	0.5786	20.35	0.5532	21.50	0.6933
CARN* [2]		26.57	0.7420	24.62	0.6226	24.79	0.5963	22.17	0.5865	21.85	0.6834
ZSSR [27]		26.45	0.7279	24.78	0.6268	24.97	0.5989	22.11	0.5805	23.53	0.7240
Pan *et al.* [24] + CARN [2]		18.10	0.4843	16.59	0.3994	18.46	0.4481	15.47	0.3872	16.78	0.5371
CARN [2] + Pan *et al.* [24]		28.69	0.8092	26.40	0.6926	26.10	0.6528	23.46	0.6597	25.84	0.8035
$\mathcal{P}$+ SFTMD		29.29	0.9014	26.40	0.7137	26.16	0.7648	22.97	0.6722	24.24	0.7950
IKC (ours)		31.52	0.9278	28.26	0.7688	27.29	0.8014	25.33	0.7760	29.90	0.8793

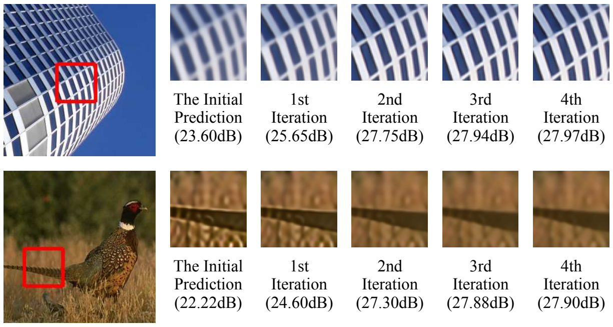


Figure 7. The intermediate SR results during kernel correction.

Table 3. Quantitative performance of the proposed IKC method on other downsampling settings.

Method	Kernel Width	BSD100 [21]		BSD100 [21]	
		PSNR	SSIM	PSNR	SSIM
CARN [2]	2.0	26.05	0.6970	25.92	0.6601
ZSSR [27]		25.64	0.6771	25.64	0.6446
CARN [2]+Pan *et al.* [24]		25.71	0.7115	25.94	0.6804
$\mathcal{P}$+ SFTMD		23.42	0.6812	25.01	0.7231
IKC, *w/o* PCA		26.85	0.7694	26.30	0.7812
IKC (ours)		27.06	0.7704	26.35	0.7838
CARN [2]	3.0	24.20	0.6066	24.53	0.5812
ZSSR [27]		24.19	0.6045	24.53	0.5796
CARN [2]+Pan *et al.* [24]		25.62	0.6678	25.52	0.6293
$\mathcal{P}$+ SFTMD		23.30	0.6799	24.41	0.7214
IKC, *w/o* PCA		26.75	0.7685	26.28	0.7849
IKC (ours)		26.98	0.7694	26.58	0.7994

proved by using the proposed IKC method. An example is shown in Figure 6. The PSNR values of different methods on different blur kernels are shown in Figure 9. As can be seen, when the kernel width becomes larger, the SR performance of the previous methods decreases. Meanwhile, the proposed IKC method achieves superior performance under all blur kernels.

To further show the generalization ability of the proposed IKC method, we test our method on another widely-used degradation setting [36], which involves Gaussian kernels and direct downsampler. When the downsampling function is different, the LR images obtained by the same blur kernel are also different. Table 3 shows the quantitative results of the proposed IKC method under different downsampling settings. The proposed IKC method has maintained its performance, which indicates that IKC is able to generalize to a downsampling setting that is inconsistent with the training settings. An important reason why the IKC method has such generalization ability is that IKC learns the kernel after PCA rather than the kernel parameterized by kernel width. PCA provides a feature representation

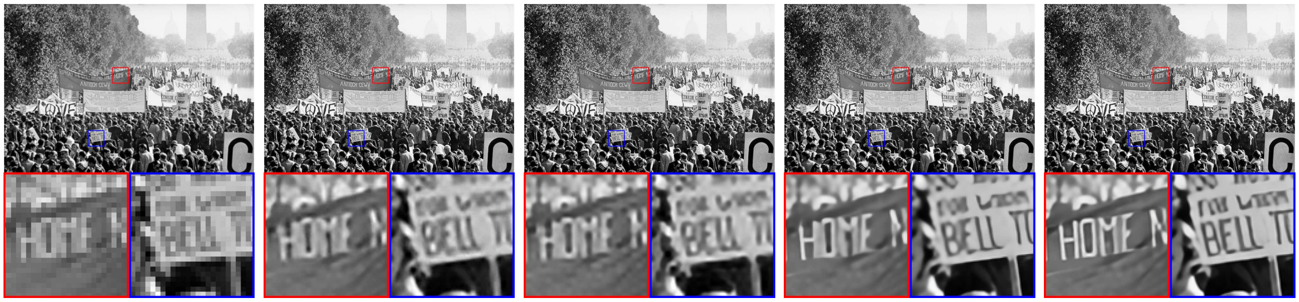

LR image A+ [31] ZSSR [27] CARN [2] IKC (ours)

Figure 8. SISR performance comparison of different methods with SR factor 4 on a real historic image '*1967 Vietnam war protest*'.

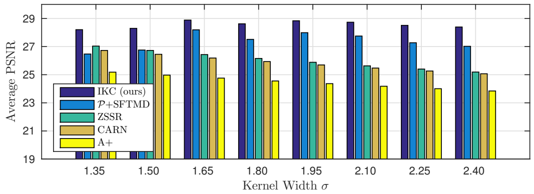


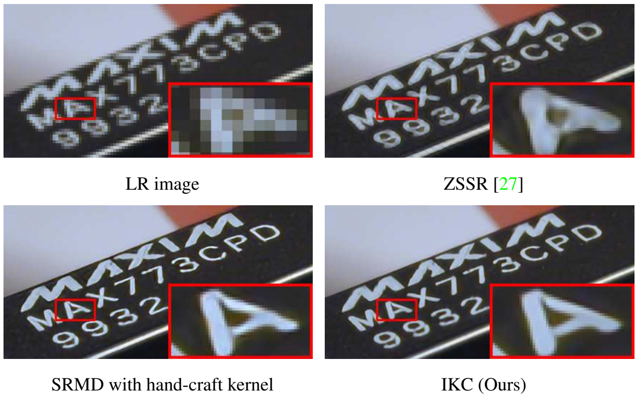
Figure 9. The PSNR performance of different methods on BSD100 [21] with different kernel width. The test SR factor is 3.

for the kernels. IKC learns the relationship between the SR images and these features rather than the Gaussian kernel width. In Table 3, we provide the comparison with the IKC method that adopts kernels parameterized by Gaussian kernel width. Experiments prove that the use of PCA helps to improve the generalization performance of IKC.

4.4. Experiments on Real Images Set

Besides the above experiments on synthetic test images, we also conduct experiments on real images to demonstrate the effectiveness of the proposed IKC and SFTMD. Since there are no ground-truth HR images, we only provide the visual comparison. Figure 8 shows the SISR results on real world image from the Historic dataset. For comparison, the A+ [31] and CARN [2] are used as the representative SR methods with bicubic downsampling, and ZSSR [27] is used as the representative blind SR method. For a real-world image, the downsampling kernel is unknown and complicated, thus performance of the non-blind SR methods are severely affected. The SOTA blind method – ZSSR also fails to provide satisfactory results. In comparison, IKC provides artifact-free SR result with sharp edges.

We also compare the proposed IKC method with the non-blind SR method using 'hand-craft' kernel on real-world image '*Chip*'. We super-resolve the LR image using SRMD with the 'hand-craft' kernel suggested by [39]. They use a grid search strategy to find the kernel parameters with good visual quality. The visual comparison is shown in Figure 10. We can see that the result of SRMD has harper edges and higher contrast, but also looks a little artificial. At the same time, IKC could provide visual pleasing SR results *automatically*. Although the contrast of IKC result is not as high as SRMD result, it still provides sharp edges and more natural visual effects.


LR image ZSSR [27]

SRMD with hand-craft kernel IKC (Ours)

Figure 10. SR results of the real image "*Chip*" with SR factor 4. The hand-craft kernel width suggested by SRMD is 1.5.

5. Discussion

In this paper, we explore the relationship between blur kernel mismatch and the SR results, then propose an iterative blind SR method – IKC. We also propose SFTMD, a new SR network architecture for multiple blur kernels. In this paper, our experiments are mainly conducted on the isotropic kernels. However, the isotropic kernels don't seem to be applicable in some real world applications. As in most cases, there are some slightly motion blurs that affect the kernel. It is worth noting that the asymmetry of the kernel mismatch effect that IKC relies on can still be observed in the case of slightly motion blur (anisotropy blur kernels). For example, the artifacts and blur of a SR image in a certain direction is related to the width of the kernel in the same direction. This indicates that, by employing such asymmetry of the kernel mismatch in each direction, the IKC method can also be applied to more realistic cases with slightly motion blur, which will be our future work.

Acknowledgements. This work is partially supported by SenseTime Group Limited, National Key Research and Development Program of China (2016YFC1400704), Shenzhen Research Program (JCYJ20170818164704758, JCYJ20150925163005055, CXB201104220032A), and Joint Lab of CAS-HK.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, volume 3, page 2, 2017. 5

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 252–268, 2018. 6, 7, 8

[3] Isabelle Begin and FR Ferrie. Blind super-resolution using a learning-based approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 85–89. IEEE, 2004. 1

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 6, 7

[5] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 185–200, 2018. 1

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. 1, 2

[7] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016. 2

[8] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. 2

[9] Netalee Efrat, Daniel Glasner, Alexander Apartsin, Boaz Nadler, and Anat Levin. Accurate blur models vs. image priors in single image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2832–2839, 2013. 1

[10] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep backprojection networks for super-resolution. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[11] He He and Wan-Chi Siu. Single image super-resolution using gaussian process regression. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 449–456. IEEE, 2011. 1

[12] Yu He, Kim-Hui Yap, Li Chen, and Lap-Pui Chau. A soft map framework for blind super-resolution image reconstruction. *Image and Vision Computing*, 27(4):364–373, 2009. 1

[13] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 7

[14] Neel Joshi, Richard Szeliski, and David J Kriegman. Psf estimation using sharp edge prediction. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 1, 2

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016. 1, 2

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate superresolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 5, 2017. 1, 2

[19] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017. 1, 2, 4

[20] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE conference on computer vision and pattern recognition (CVPR) workshops*, volume 1, page 4, 2017. 2

[21] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001. 6, 7, 8

[22] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 7

[23] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–952, 2013. 1, 2, 3

[24] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Deblurring images via dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2315–2328, 2018. 6, 7

[25] Gernot Riegler, Samuel Schulter, Matthias Ruther, and Horst Bischof. Conditioned regression models for non-blind single image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 522–530, 2015. 1, 2, 6

[26] 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

[27] Assaf Shocher, Nadav Cohen, and Michal Irani. Zero-shot super-resolution using deep internal learning. In *Conference on computer vision and pattern recognition (CVPR)*, 2018. 1, 2, 6, 7, 8

[28] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 5, 2017. 2

[29] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2017. 1

[30] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1110–1121. IEEE, 2017. 5

[31] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014. 7, 8

[32] Qiang Wang, Xiaoou Tang, and Harry Shum. Patch based blind image super resolution. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 709–716. IEEE, 2005. 1

[33] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 606–615, 2018. 2, 4, 5

[34] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *European Conference on Computer Vision*, pages 63–79. Springer, 2018. 1, 2

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[36] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *European Conference on Computer Vision*, pages 372–386. Springer, 2014. 1, 2, 7

[37] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2018. 2, 3

[38] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 6, 7

[39] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 6, 2018. 1, 2, 3, 4, 6, 8

[40] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018. 1, 2

[41] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2