Mutual Affine Network for Spatially Variant Kernel Estimation in Blind Image Super-Resolution

Jingyun Liang[1]   Guolei Sun[1]   Kai Zhang[1,*]   Luc Van Gool[1,2]   Radu Timofte[1]

[1]Computer Vision Lab, ETH Zurich, Switzerland   [2] KU Leuven, Belgium

{jinliang, guosun, kai.zhang, vangool, timofter}@vision.ee.ethz.ch

https://github.com/JingyunLiang/MANet

Abstract

Existing blind image super-resolution (SR) methods mostly assume blur kernels are spatially invariant across the whole image. However, such an assumption is rarely applicable for real images whose blur kernels are usually spatially variant due to factors such as object motion and out-of-focus. Hence, existing blind SR methods would inevitably give rise to poor performance in real applications. To address this issue, this paper proposes a mutual affine network (MANet) for spatially variant kernel estimation. Specifically, MANet has two distinctive features. First, it has a moderate receptive field so as to keep the locality of degradation. Second, it involves a new mutual affine convolution (MAConv) layer that enhances feature expressiveness without increasing receptive field, model size and computation burden. This is made possible through exploiting channel interdependence, which applies each channel split with an affine transformation module whose input are the rest channel splits. Extensive experiments on synthetic and real images show that the proposed MANet not only performs favorably for both spatially variant and invariant kernel estimation, but also leads to state-of-the-art blind SR performance when combined with non-blind SR methods.

1. Introduction

Single image super-resolution (SR), with the aim of reconstructing the high-resolution (HR) image from a low-resolution (LR) image, is a classical vision problem. Recently, convolutional neural networks (CNNs) [42, 40, 39, 7, 5, 21, 15, 44, 24, 23] have been widely used in SR. However, most of these SR methods assume the blur kernel is ideal and fixed (usually a bicubic kernel), and thus deteriorate seriously if the real kernel deviates from the ideal one [46, 3, 53, 14]. Therefore, dealing with unknown blur kernels, *i.e.*, blind SR, is becoming a hot topic.

---

*Corresponding author.

Figure 1: Kernel estimation results of the proposed MANet on "*img017*" in Urban100 [19] for scale factor 4. The shown image is the SR image, whose corresponding HR image was blurred by a spatially invariant kernel as shown in the top right green rectangle.

While existing blind SR methods [14, 3, 54, 53, 32, 25] have achieved remarkable performance, they assume blur kernels are spatially invariant and only estimate a single kernel for the whole image, giving rise to two inherent problems. First, real-world blur kernels are typically spatially variant. Due to different environmental factors like object motion and depth difference, as well as non-ideal imaging such as out-of-focus and camera shake [37, 2], blur kernels at different locations of the image tend to be different. Second, estimating a single kernel for the whole image is susceptible to the adverse effects of flat patches, even under the spatially invariant assumption. For a natural image, some patches contain edges or corners that are discriminative for kernel estimation (*e.g.*, the pillars in Fig. 1), while some other patches are rather flat (*e.g.*, the blue sky) and are less discriminative since they correspond to various indistinguishable but correct blur kernels that all result in the same LR patch. Therefore, estimating spatially variant kernels is more reasonable for blind SR.

The main challenge of spatially variant kernel estimation lies in the locality of degradation. A blur kernel only has impacts on a local image patch of the same size, *e.g.*, 21×21, which becomes even smaller after downsampling (*e.g.*, about 5×5 when scale factor is 4). Furthermore, utilizing pixels outside of the impacted patch may be detrimen-

tal when nearby kernels are different, as shown in Fig. 4(b). Therefore, an ideal kernel estimation model should estimate kernel from the impacted image patch. This is very challenging due to the ill-posedness of the problem. An appealing option is CNN, which has shown great promise for ill-posed problems [8, 52, 50]. However, most of existing networks have very large receptive fields, making them unsuitable for kernel estimation.

To tackle the problem, we propose the **M**utual **A**ffine **N**etwork (MANet) that has a moderate receptive field. More specifically, MANet consists of feature extraction and kernel reconstruction module. The first module uses several residual blocks, along with downsampler layer, upsampler layer and skip connections, to extract image feature from the LR image input, while the second module reconstructs kernels for every HR image pixel from the feature. In particular, we propose the mutual affine convolution (MAConv) layer for the residual block, in order to exploit channel interdependence without increasing network receptive field. It splits a feature along the channel dimension, and then transforms each split by the affine transformation module whose parameters are learned from the rest splits. After that, each split is fed into a convolution layer and then concatenated as the MAConv layer output.

The main contributions of this work are as follows:

- We propose a kernel estimation framework named MANet. With a moderate receptive field (*i.e.*, 22×22), it estimates kernels from tiny LR image patches. The minimum patch from which it can accurately estimate a kernel is of size 9×9.

- We propose the mutual affine convolution layer to enhance feature expressiveness by exploiting channel interdependence without increasing network receptive field, making it suitable for feature extraction of blur kernels. It also reduces model parameters and computation cost by about 30% compared with plain convolution layer.

- Compared with existing methods, MANet performs favourably for both spatially variant and invariant kernel estimation, leading to state-of-the-art blind SR performance when combined with non-blind SR models. It also shows good properties in dealing with different kinds of patches, *e.g.*, estimating kernels accurately from non-flat patches and producing fixed kernels for flat patches.

2. Related Work

Kernel estimation. Prior to the deep learning era, blind SR methods typically estimate HR image and kernel via image patch or edge prior information [16, 34, 12, 55, 29]. Recently, several attempts have been made on using deep neural networks for blind SR. Bell-Kligler *et al*. [3] propose KernelGAN which trains an internal generative adversarial network (GAN) on a single image. Based on KernelGAN, Liang *et al*. [25] propose KernelGAN-FKP to incorporate

a flow-based kernel prior into the framework. Nevertheless, KernelGAN and its variant are not suitable for low-resolution images and large scale factors. The GAN optimization also brings unstable estimation and long testing time. Cornillere *et al*. [6] propose SRSVD to use a kernel discriminator to evaluate the non-blind SR model output and optimize kernel latent variables by minimizing the discriminator error. However, it needs to optimize kernels patch by patch for spatially variant SR, which is inefficient and ineffective. Gu *et al*. [14] propose IKC for kernel estimation on the basis of paired training data. They first estimate the PCA feature of kernel by a CNN network and then iteratively correct it by alternating optimization. One common problem of SRSVD and IKC is that they predict the kernel feature rather than the kernel itself, limiting their combination with other methods.

Non-blind SR. Non-blind SR models aim to reconstruct the HR image given estimated kernels. Gernot *et al*. [32] transform SRCNN [8] to a non-blind model by replacing the weight of the first convolution layer with the kernel feature. Zhang *et al*. [53] propose a stretching strategy to take kernels as additional input and train an end-to-end model SRMD. Based on SRMD, Gu *et al*. [14] propose the SFTMD model that inputs kernels by the SFT layer [41], while Xu *et al*. [45] incorporate dynamic convolution into their UDVD model. Besides, Zhang *et al*. [49] decompose SR into deblurring and denoising, and use a trainable neural network to solve these two sub-problems iteratively. Different from above methods, Shocher *et al*. [35] propose zero-shot models to train image-specific networks at test time based on patch recurrence property. It is worth pointing out that the proposed MANet focuses on kernel estimation and could be combined with most non-blind models.

Other methods. There are other related methods that do not explicitly estimate blur kernels, such as unpaired SR [47, 26, 13, 27, 22, 43], zero-shot SR [35, 36, 20] and data augmentation techniques [51]. However, these methods often suffer from pixel misalignment problem and are hard to be compared quantitatively.

3. Methodology

3.1. Problem Formulation

Mathematically, the LR image $\boldsymbol{I}^{LR}$ is generated from the HR image $\boldsymbol{I}^{HR}$ by a degradation model. When blur kernels are spatially invariant [10, 11], the relation between $\boldsymbol{I}^{LR}$ and $\boldsymbol{I}^{HR}$ is modelled as

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

where $\otimes$ denotes the convolution between $\boldsymbol{I}^{HR}$ and blur kernel $\boldsymbol{k}$, $\downarrow_s$ represents downsampling with scale factor s, and $\boldsymbol{n}$ is noise. For blind SR, both HR image $\boldsymbol{I}^{HR}$ and blur
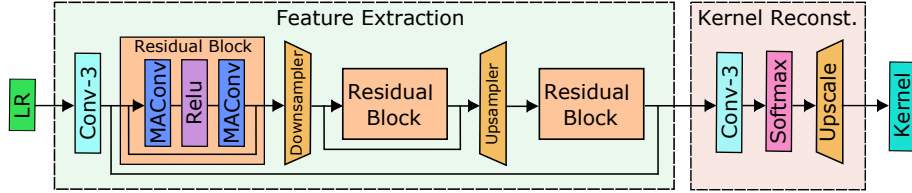
Figure 2: Architecture of the proposed mutual affine network (MANet). Given a LR image input $\boldsymbol{I}^{LR} \in \mathcal{R}^{C \times \frac{H}{s} \times \frac{W}{s}}$, the network outputs the kernel estimation $\boldsymbol{K} \in \mathcal{R}^{hw \times H \times W}$ for the corresponding HR image $\boldsymbol{I}^{HR} \in \mathcal{R}^{C \times H \times W}$. MANet is composed of feature extraction and kernel reconstruction modules. In particular, in the feature extraction module, each residual block consists of two mutual affine convolution (MAConv) layers, while the Downsampler, Upsampler and Upscale blocks are implemented by 2×2 convolution (stride of 2), 2×2 transpose convolution (stride of 2) and nearest neighbor interpolation (scale factor of s), respectively.

kernel $\boldsymbol{k}$ are unknown. This problem is very ill-posed since many different pairs of $\boldsymbol{I}^{HR}$ and $\boldsymbol{k}$ can give rise to the same LR image $\boldsymbol{I}^{LR}$. When blur kernels are spatially variant, the problem becomes even more ill-posed. In this case, images and kernels can be written in vector forms. The degradation process is modelled as

$$\boldsymbol{I}^{LR} = (\boldsymbol{K}\boldsymbol{I}^{HR}) \downarrow_s + \boldsymbol{n}, \qquad (2)$$

where $\boldsymbol{K}$ denotes the blur matrix similar to the convolution matrix. Since the i-th row of $\boldsymbol{K}$ corresponds to the blur kernel of the i-th pixel in $\boldsymbol{I}^{HR}$, $\boldsymbol{K}$ is no longer a Toepliz matrix as in spatially invariant degradation,

In the literature of blind SR [3, 34, 14, 53, 46, 32, 16, 31, 35, 45, 51], kernels are widely assumed to be Gaussian as many estimated real-world kernels (*e.g.*, [29, 34]) are actually unimodal and can typically be modeled by a Gaussian [32, 9, 46]. Besides, Gaussian kernel is reasonable and challenging for the SR problem (in contrast to image deblurring). It is also practical for quantitative evaluation. Therefore, following the practice of spatially invariant blind SR, we adopt this assumption for the more challenging spatially variant blind SR as well. Note that our model is learning-based and not restricted to specific kernel assumptions. If other kernel distributions prove to be more reasonable, it is easy to re-train our model.

3.2. Proposed Method

As observed in [12, 55, 29], image patches blurred by different kernels have different patch distributions. Kernel-GAN [3] exploits this property by an internal GAN and uses a discriminator to discriminate image patches as real or fake. However, it only works for spatially invariant kernel estimation and cannot estimate kernels for tiny image patches. To take one step further, we propose to estimate kernels directly from image patches.

Overall framework. Modern neural networks often stacks multiple layers to build deep models with large receptive fields. However, for the task of spatially variant kernel estimation, we need to keep the locality of degradation. Hence, we propose a mutual affine network (MANet) with a moderate receptive field.

More specifically, as shown in Fig. 2, MANet contains two modules: feature extraction and kernel reconstruction modules. Inspired by U-Net [33], feature extraction module is composed of convolution layers, residual blocks, a downsampler and an upsampler. The LR image is first input to a 3×3 convolutional layer to extract image feature, which then goes through 3 residual blocks. Each residual block contains two proposed mutual affine convolution layers with ReLU activation between them for learning nonlinearity. Before and after the intermediate residual block, a convolution and a transpose convolution layer (both with stride of 2) are used for downsampling and upsampling the image feature, respectively. Additionally, we add two skip connections in feature extraction module to utilize different levels of features and improve representation capability.

After feature extraction, kernel reconstruction module uses a 3×3 convolution layer and a softmax layer along the channel to predict the kernels for every LR image pixel. Then, we use nearest neighbor interpolation to obtain the final kernel predictions for the HR image. With a slight abuse of notation, the kernel prediction is denoted as $\boldsymbol{K} \in \mathcal{R}^{hw \times H \times W}$, where h, w, H and W are kernel height, kernel width, HR image height and HR image width, respectively.

With elaborate architecture design, MANet has a moderate receptive field of 22×22 on the LR image input, which ensures that kernel estimation would not be interfered by other image patches farther than 11 pixels. Meanwhile, it has enough capability to predict kernels with the mutual affine convolution layer as to be described below.

Mutual affine convolution. Generally, small receptive field means shallow networks, which have less representation capacity to learn kernels from various image patches. One possible solution is to increase the channel number. However, it brings quadratic increases of parameters and computation burden. Instead, we propose a novel mutual affine convolution (MAConv) layer to solve the problem.

Let $\boldsymbol{x} \in \mathcal{R}^{C_{in} \times H_f \times W_f}$ be the input feature of the MAConv layer. As shown in Fig. 3, we first divide $\boldsymbol{x}$ into S splits along the channel as

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_S = \mathrm{split}(\boldsymbol{x}). \qquad (3)$$
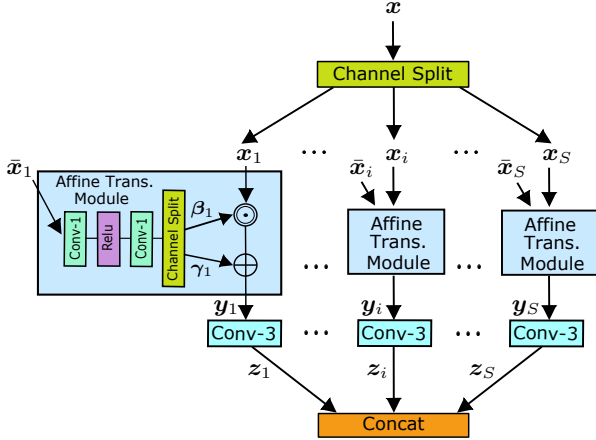
Figure 3: Illustration of the mutual affine convolution (MAConv).

For each split $\boldsymbol{x}_i \in \mathcal{R}^{\frac{C_{in}}{S} \times H_f \times W_f}$, we denote the concatenation of splits that are complementary to $\boldsymbol{x}_i$ as $\bar{\boldsymbol{x}}_i \in \mathcal{R}^{\frac{C_{in}(S-1)}{S} \times H_f \times W_f}$. Both $\boldsymbol{x}_i$ and $\bar{\boldsymbol{x}}_i$ are passed into the affine transformation module, which has a fully-connected network $\mathcal{F}$ to learn transformation parameters $\boldsymbol{\beta}_i$ and $\boldsymbol{\gamma}_i$ from $\bar{\boldsymbol{x}}_i$. Then, $\boldsymbol{\beta}_i$ and $\boldsymbol{\gamma}_i$ are used to scale and shift $\boldsymbol{x}_i$, respectively. The whole process is formulated as

$$\begin{aligned}\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i &= \text{split}(\mathcal{F}(\bar{\boldsymbol{x}}_i)), \\ \boldsymbol{y}_i &= \boldsymbol{\beta}_i \odot \boldsymbol{x}_i + \boldsymbol{\gamma}_i,\end{aligned} \quad (4)$$

where $\odot$ denotes the Hadamard product. $\mathcal{F}$ is composed of two 1×1 convolution layers and an in-between ReLU activation layer. The input, hidden and output channels are set to $\frac{C_{in}(S-1)}{S}$, $\frac{C_{in}(S-1)}{2S}$ and $\frac{2C_{in}}{S}$, respectively.

After transformation, for all $i \in \{1, 2, ..., S\}$, we use a 3×3 convolution layer to generate feature $\boldsymbol{z}_i = \text{conv}_i(\boldsymbol{y}_i)$. $\boldsymbol{z}_i \in \mathcal{R}^{\frac{C_{out}}{S} \times H_f \times W_f}$ when MAConv has C_{out} output channels. Finally, different features $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_S$ are concatenated to generate the MAConv output

$$\boldsymbol{z} = \text{concat}(\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_S). \quad (5)$$

MAConv exploits the interdependence between different channels by mutual affine transformation, instead of fully connecting all input and output channels as in plain convolution layer. Such a design can improve feature representation capacity and largely reduce model size as well as computation complexity. For plain 3×3 convolution, number of parameters and floating point operations (FLOPs) are $9C_{in}C_{out}$ and $9C_{in}C_{out}H_fW_f$, respectively. In contrast, MAConv only has $\frac{9}{S}C_{in}C_{out} + \frac{S^2-1}{2S}C_{in}^2$ parameters and $(\frac{9}{S}C_{in}C_{out} + \frac{2(S-1)}{S^2}C_{in}^2)H_fW_f$ FLOPs, which are much smaller when choosing proper S. Comparisons of exact parameters and FLOPS are shown in Table 1.

It is noteworthy that the receptive field of MAConv is still the same as a single 3×3 convolution layer, as the affine transformation modules do not increase receptive field. In comparison, popular feature extraction blocks such as dense block [18] and squeeze-and-excitation (SE) block [17] lead to tremendous increase of receptive field and thus are not suitable for kernel estimation networks.

Loss function. Mean absolute error (MAE) is used as the loss function to measure the difference between estimated kernels and ground-truth kernels. Specifically, the loss function is

$$L = \frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{i=1}^{H} \sum_{j=1}^{W} \|\boldsymbol{K}_{ij}^{(n)} - \boldsymbol{G}_{ij}^{(n)}\|_1, \quad (6)$$

where $\boldsymbol{K}_{ij}^{(n)}$ and $\boldsymbol{G}_{ij}^{(n)}$ denote the estimated kernel and the corresponding ground-truth at position (i, j) on training sample n. N, H and W are the total number, height and width of training samples, respectively.

4. Experiments

4.1. Experimental Setup

Implementation details. Following existing blind SR works [3, 34, 14, 53, 46, 32, 16, 31, 35, 45, 51], we conduct experiments on 21×21 anisotropic Gaussian kernels. In training, kernel widths $\sigma_1, \sigma_2 \sim \mathcal{U}(0.175s, 2.5s)$ for scale factor s, while rotation angle $\theta \sim \mathcal{U}(0, \pi)$. We randomly crop 192×192 image patches from DIV2K [1] and augment them by random flip and rotation. Then, image patches are blurred by random kernels. It is worth pointing out that the network can learn to deal with spatially variant kernels even trained on spatially invariant blurred images. For MANet, channel numbers of three residual blocks are set to 128, 256 and 128, respectively. Channel split number S is 2 unless specified. For non-blind SR, we first train a modified RRDB-SFT network with 10 RRDB blocks [42] and SFT layers [41]. Then, we fine-tune RRDB-SFT on kernels estimated by MANet. Details on training procedure and RRDB-SFT architecture are provided in the supplementary.

Performance evaluation. In our ablation study and spatially invariant experiments, we sample kernels in an evenly spaced manner: $\sigma_1, \sigma_2 \in \{1, 5, 9\}$ and $\theta \in \{0, \frac{\pi}{4}\}$ when scale factor is 4. For scale factors 2 and 3, we keep the same procedure and sample kernel widths from $\{1, 3, 5\}$ and $\{1, 4, 7\}$, respectively. This sampling strategy means that, after kernel deduplication, every image in testing sets is degraded by 9 different kernels, resulting in 9 testing pairs. Separately, kernel sampling details for spatially variant experiments are given in Table 3. For kernel evaluation, it is not suitable to use kernel PSNR since an image patch may correspond to multiple correct kernels. Hence, we use reconstructed LR image PSNR/SSIM for evaluation. For image evaluation, we compare SR image PSNR/SSIM on the Y channel of YCbCr space.

Table 1: Comparison of plain convolution, group convolution and MAConv. '#Channel' represents channel numbers of residual blocks in MANet, while '#Split' represents group numbers for group convolution or channel split number for MAConv. LR image PSNR/SSIM are tested on BSD100 [28] for scale factor 4. '#Params', 'Memory', 'FLOPs' and 'Runtime' are tested on a 256×256 LR image input.

Type	#Channel	#Split	LR Image PSNR/SSIM	#Params [M]	Memory [M]	FLOPs [G]	Runtime [s]
Plain Convolution	[32, 64, 32]	-	46.22/0.9951	0.2557	234.229	12.6804	0.006499
	[64, 128, 64]		47.65/0.9965	0.7649	248.170	33.9508	0.010534
	[128, 256, 128]		48.85/0.9974	2.5451	278.959	102.2613	0.018363
Group Convolution	[32, 64, 32]	2	45.14/0.9930	0.2004	234.018	10.8685	0.006490
	[64, 128, 64]	2	46.72/0.9957	0.5437	247.326	26.7030	0.009985
	[128, 256, 128]	2	48.32/0.9969	1.6603	275.584	73.2703	0.012948
	[128, 256, 128]	4	47.98/0.9967	1.2180	273.896	58.7748	0.012562
	[128, 256, 128]	6	47.74/0.9965	1.0452	272.991	52.7863	0.012382
MAConv (ours)	[32, 64, 32]	2	45.87/0.9946	0.2102	234.068	11.1757	0.011263
	[64, 128, 64]	2	47.74/0/9965	0.5818	247.481	27.9215	0.016451
	[128, 256, 128]	2	49.39/0.9978	1.8104	276.162	78.1231	0.020956
	[128, 256, 128]	4	49.77/0.9979	1.5902	275.334	70.9173	0.025172
	[128, 256, 128]	6	49.80/0.9979	1.6451	275.596	72.6990	0.030595

(a) #MAConv=2, kernel loss　　　　(b) #MAConv=4, kernel loss　　　　(c) #MAConv=2, LR image loss
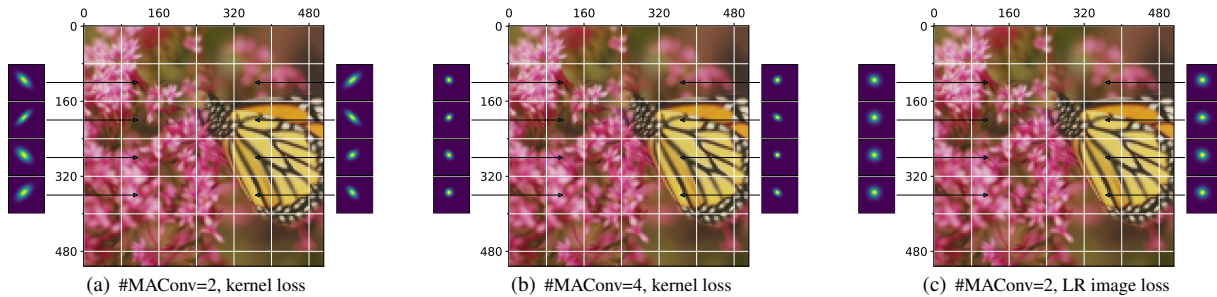
Figure 4: Comparison of different numbers of MAConv layers in a residual block and different training losses when scale factor is 4. The shown images are nearest neighbour interpolations of the LR image, whose corresponding HR image was divided into patches of size 80×80. For all image patches, Gaussian kernel parameters $\sigma_1 = 10$ and $\sigma_2 = 0.7$. In particular, for different patches, θ is set to $\frac{\pi}{4}$ when the sum of spatial patch coordinates is even, and $\frac{3\pi}{4}$ otherwise.

4.2. Ablation Study

MAConv vs. other convolutions. Comparison among plain convolution, group convolution and the proposed MAConv are shown in Table 1, from which we have the following observations. First, MAConv achieves best performance on LR image PSNR/SSIM, indicating that its resulting kernels could better preserve data fidelity compared with its competitors. It also has significantly less parameters and FLOPs than plain convolution. Note that, unlike FLOPs, the runtime of MAConv is slightly longer than plain convolution because the implementation code is not optimized for parallel computing of different splits. Second, with the increase of the channel number, the kernel estimation performance of MAConv is improved, accompanying with number of parameters and FLOPs rising up. Third, kernel estimation performance of MAConv has an increasing tendency with the number of splits. This implies that larger number of splits can better exploit channel interdependence and increase feature representation capability. To balance accuracy and runtime, we set channel numbers and split number to [128, 256, 128] and 2, respectively.

Different numbers of MAConv layers. We increase the MAConv layer number in a residual block from 2 to 4 to explore its effects on kernel estimation. Accordingly, the receptive field of MANet is increased from 22×22 to 38×38. As shown in Figs. 4(a) and 4(b), on a toy example image

whose neighboring patches have different kernels, MANet with 2 MAConv layers can estimate kernels for different patches accurately, but it fails to generate accurate kernel estimations when MAConv layer number is 4. This actually accords with our previous analysis: when the model has a large receptive field and takes pixels far from the center pixel into account for kernel estimation, its results may be affected by other image patches. This is not a desired property for spatially variant kernel estimation.

Kernel loss vs. LR image loss. Another choice of loss function is the LR image loss, which corresponds to the data fidelity term in the Maximum A Posteriori (MAP) framework. It is defined as the mean absolute error (MAE) between the LR image and the corresponding LR image reconstruction. As a relaxation of kernel loss, LR image loss only requires that the kernel can reconstruct the LR image with high fidelity. Figs. 4(a) and 4(c) show the comparison between kernel loss and LR image loss. As one can see, MANet succeeds to estimate kernels when trained with kernel loss. However, when using LR image loss, MANet cannot discriminate different kinds of image patches and always predict a fixed kernel, which could be the average of all possible kernels. Note that, even MANet is forced to estimate kernels accurately from all kinds of patches with the kernel loss, it learns to estimate kernels accurately from non-flat patches and generates fixed kernels for flat patches.
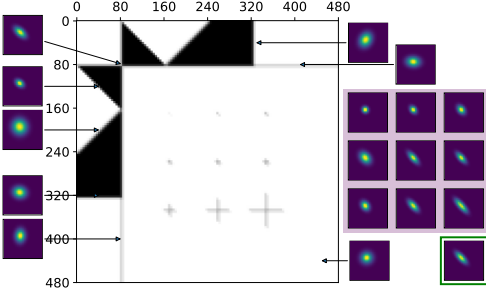
Figure 5: Kernel estimation results of MANet at different positions on a synthetic image when scale factor is 4. The image is generated by blurring with a Gaussian kernel with $\sigma_1 = 6$, $\sigma_2 = 1$ and $\theta = \frac{\pi}{4}$, as shown in the down right green rectangle. The HR image (can be found in the supplementary) has 9 black crosses (1×1, 3×3, 5×5, 7×7, 9×9, 11×11, 21×21, 41×41 and 61×61), whose kernel predictions are shown in the right purple rectangle. **Best viewed by zooming**.

Table 2: Kernel estimation results under complex noise corruption. σ and q denote Gaussian noise level and JPEG compression level, respectively. The reported LR image PSNR/SSIM is tested on BSD100 [28] for scale factor 4.

σ \ q	70	80	90	100
0	43.37/0.9918	44.13/0.9929	44.82/0.9940	45.45/0.9947
5	43.23/0.9917	43.67/0.9924	43.88/0.9928	44.54/0.9938
10	42.33/0.9902	42.43/0.9904	43.16/0.9916	43.66/0.9925
15	40.59/0.9849	40.81/0.9865	41.36/0.9872	42.56/0.9905

4.3. Experiments on Kernel Estimation

We plot kernel estimation results on a testing image in Fig. 1. As we can see, MANet can accurately estimate kernels from non-flat patches (*e.g.*, the pillars) and tends to predict a fixed kernel for flat patches (*e.g.*, the blur sky), which could be the average of all possible kernels. The kernels are diversified and may not be identical to the ground-truth kernel, but most of them are "correct" kernels, as indicated by the high LR image PSNR (the data fidelity). Visualization of kernel distribution is provided in the supplementary. We also test MANet on a synthetic image for better understanding. As shown in Fig. 5, MANet can estimate kernels accurately from a minimum image patch of size 9×9. The performance is further improved when patch size is increased. When there is no corner (only edges) in a small patch, MANet cannot estimate kernels accurately due to insufficient information. For flat patches without corners and edges, MANet would estimate a fixed isotropic-like kernel.

In real-world scenarios, images may suffer from noise corruption or compression artifacts. To test kernel estimation performance in more complex cases, we add Gaussian and JPEG compression noises during training, and test it on different noise levels and compression levels. As shown in Table 2, even though there is a performance drop compared with the noisy-free case, the LR image PSNR ranges from 40.59 to 45.45dB, which shows the potential to estimate kernels under heavy noisy corruptions.

4.4. Experiments on Spatially Variant SR

We compare MANet with baseline models and existing blind SR models: HAN [30], DIP [38], KernelGAN [3], HAN with correction [20], SRSVD [6], IKC [14] (retrained with anisotropic Gaussian kernel and the same non-blind SR model RRDB-SFT as MANet) and the upper bound model (RRDB-SFT given ground-truth kernels). As shown in Table 3, MANet leads to the best performance for different spatially variant kernel types. In particular, representative bicubic SR models RCAN and HAN suffer from severe performance drop when kernels deviate from the assumed bicubic kernel. Similarly, DIP produces unfavorable results since it also assumes that kernels are fixed. KernelGAN designs an internal GAN framework based on patch dissimilarity, but its kernel estimation performance is limited, leading to inferior SR results. SRSVD has the potential to deal with spatially variant SR by optimizing kernels patch by patch, but it significantly increases the runtime. IKC performs better than above models by learning to predict kernel directly from LR images. However, it only estimates one kernel for the whole image and has limited performance for spatially variant degradation. In comparison, the proposed MANet estimates kernels for every position on the image. Therefore, it can deal with spatially variant degradation and outperform IKC by large margins based on the same non-blind model. Even with image noises, MANet still achieves superior performance compared with other models.

Fig. 6 compares visual results of different methods. Though it is known that GAN loss can improve the visual quality, we train all these model with only L_1 pixel loss for simple and fair comparison. One can see that HAN tends to generate blurry results when kernel mismatches, whereas DIP generates images with some noise-like artifacts. SRSVD is not compared as the codes and models for scale factor 4 are not available. The kernel estimations of KernelGAN and IKC are either too smooth, or too sharp, resulting in ringing or blurry artifacts on final SR images. By comparison, our MANet is able to handle spatially variant degradation and produces the most visually pleasant results.

For the runtime and memory usage, the proposed MANet takes about 0.02 seconds and 0.3GB memory to predict kernels for a 256×256 LR image input on a Tesla V100 GPU. By contrast, KernelGAN needs about 93 seconds and consumes 1.3GB memory, while the runtime and memory usage of IKC are about 15.2 seconds and 2.0GB, respectively.

4.5. Experiments on Spatially Invariant SR

Most existing blind SR models assume blind SR has spatially invariant kernels, which is a special case of spatially variant SR. As one can see from Table 4, the proposed MANet maintains its performance and produces best results across different datasets and scale factors. Particularly, although KernelGAN can estimate kernels from LR

Table 3: Average PSNR/SSIM of different methods for **spatially variant** blind SR on BSD100 [28]. Every testing image is divided into $m \times n$ patches (patch size is 40×40), which are degraded by different kernels. According to experimental setup in Sec 4.1, for scale factor s, the Gaussian kernel width range a and minimum kernel width b are $2.325s$ and $0.175s$, respectively. In particular, for patch (i, j), its corresponding kernel is determined by a, b, $x = \frac{i}{m}$ and $y = \frac{j}{n}$, as shown in the table header. The best and second best results are highlighted in red and blue colors, respectively.

Method	Scale Factor	Noise Level	Spatially Variant Kernel Type				
			1 $\sigma_1 = a + b$ $\sigma_2 = ax + b$ $\theta = 0$	2 $\sigma_1 = ay + b$ $\sigma_2 = ax + b$ $\theta = 0$	3 $\sigma_1 = a + b$ $\sigma_2 = b$ $\theta = \pi x$	4 $\sigma_1 = ay + b$ $\sigma_2 = ax + b$ $\theta = \pi x$	5 $\sigma_1 \sim \mathcal{U}(b, a+b)$ $\sigma_2 \sim \mathcal{U}(b, a+b)$ $\theta \sim \mathcal{U}(0, \pi)$
HAN [30]	×2	0	24.98/0.6424	25.31/0.6721	25.04/0.6593	25.41/0.6801	25.19/0.6643
DIP [38]	×2	0	26.11/0.6765	25.04/0.6693	23.81/0.6375	25.05/0.6695	25.40/0.6780
KernelGAN [3]	×2	0	24.81/0.6579	23.68/0.6391	21.88/0.5456	23.63/0.6410	23.49/0.6309
HAN [30] + Correction [20]	×2	0	27.88/0.7634	27.18/0.7352	25.53/0.6996	26.25/0.7101	25.88/0.6811
SRSVD [6]	×2	0	28.53/0.8019	27.81/0.7871	27.44/0.7819	27.81/0.7765	27.81/0.7788
RRDB-SFT + IKC [14]	×2	0	28.45/0.7996	27.92/0.7922	27.49/0.7854	27.83/0.7946	27.75/0.7755
RRDB-SFT + MANet (ours)	×2	0	30.09/0.8397	30.70/0.8610	29.15/0.8305	30.46/0.8567	28.27/0.7957
RRDB-SFT + GT (upper bound)	×2	0	30.71/0.8578	31.47/0.8809	29.63/0.8582	31.32/0.8804	28.37/0.8460
HAN [30]	×3	0	23.29/0.5591	23.22/0.5713	23.03/0.5537	23.17/0.5641	23.08/0.5603
DIP [38]	×3	0	25.75/0.6507	25.38/0.6573	23.75/0.6105	25.32/0.6583	25.71/0.6660
RRDB-SFT + IKC [14]	×3	0	27.07/0/7357	26.86/0.7352	26.31/0.7188	26.87/0.7377	26.71/0.7189
RRDB-SFT + MANet (ours)	×3	0	28.48/0.7753	28.51/0.7780	27.72/0.7641	28.48/0.7792	26.93/0.7268
RRDB-SFT + GT (upper bound)	×3	0	28.83/0.7892	29.05/0.8011	28.27/0.7838	29.01/0.8004	27.96/0.7836
HAN [30]	×4	0	22.19/0.5111	21.83/0.5066	21.66/0.4989	22.04/0.5233	21.99/0.5136
DIP [38]	×4	0	25.24/0.6174	25.30/0.6242	24.01/0.5813	25.24/0.6229	25.31/0.6266
KernelGAN [3]	×4	0	19.90/0.4317	18.32/0.3697	17.62/0.3517	18.56/0.3826	19.02/0.3888
HAN [30] + Correction [20]	×4	0	25.13/0.6151	25.51/0.6156	24.41/0.6017	25.67/0.6454	25.82/0.6435
RRDB-SFT + IKC [14]	×4	0	26.46/0.6952	26.03/0.6880	25.58/0.6759	26.09/0.6887	26.01/0.6775
RRDB-SFT + MANet (ours)	×4	0	27.24/0.7169	27.21/0.7169	26.61/0.7070	27.16/0.7157	26.16/0.6790
RRDB-SFT + GT (upper bound)	×4	0	27.51/0.7300	27.57/0.7355	27.05/0.7227	27.53/0.7345	27.13/0.7262
HAN [30]	×4	15	20.58/0.3148	20.28/0.3078	20.53/0.3139	20.97/0.3286	20.34/0.3146
DIP [38]	×4	15	18.15/0.1854	18.14/0.2042	17.71/0.1960	18.02/0.1997	18.10/0.1998
KernelGAN [3]	×4	15	15.16/0.0992	14.68/0.0961	14.51/0.0873	15.11/0.1086	14.66/0.0859
HAN [30] + Correction [20]	×4	15	18.13/0.1840	18.21/0.2141	18.04/0.2450	18.32/0.2209	18.41/0.2281
RRDB-SFT + IKC [14]	×4	15	24.64/0.5950	24.94/0.6162	24.81/0.6175	25.01/0.6174	24.95/0.6078
RRDB-SFT + MANet (ours)	×4	15	24.89/0.6030	25.21/0.6192	25.11/0.6197	25.24/0.6200	25.05/0.6118
RRDB-SFT + GT (upper bound)	×4	15	24.98/0.6082	25.32/0.6255	25.33/0.6292	25.34/0.6264	25.30/0.6233

Table 4: Average PSNR/SSIM of different methods for **spatially invariant** blind SR on different datasets. Note that KernelGAN is not applicable to small images or large scale factors for some datasets. The best and second best results are highlighted in red and blue colors, respectively.

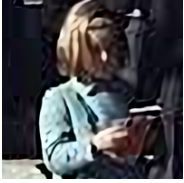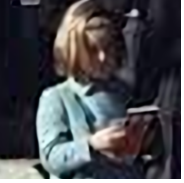| Method	Scale Factor	Noise Level	Set5 [4]	Set14 [48]	BSD100 [28]	Urban100 [19]
HAN [30]	×2	0	26.83/0.7919	23.21/0.6888	25.11/0.6613	22.42/0.6571
DIP [38]	×2	0	28.19/0.7939	25.66/0.6999	25.03/0.6762	22.97/0.6737
KernelGAN [3]	×2	0	-	23.92/0.6898	25.28/0.6395	21.97/0.6582
HAN [30] + Correction [20]	×2	0	28.61/0.8013	26.22/0.7292	26.88/0.7116	25.31/0.7109
SRSVD [6]	×2	0	34.51/0.8787	31.10.0.8581	29.71/0.7993	28.08/0.7965
RRDB-SFT + IKC [14]	×2	0	35.30/0.9381	31.48/0.8797	30.50/0.8545	28.62/0.8689
RRDB-SFT + MANet (ours)	×2	0	35.98/0.9420	31.95/0.8845	30.97/0.8650	29.87/0.8877
RRDB-SFT + GT (upper bound)	×2	0	36.64/0.9473	32.85/0.8964	31.40/0.8754	30.95/0.9069
HAN [30]	×3	0	23.71/0.6171	22.31/0.5878	23.21/0.5653	20.34/0.5311
DIP [38]	×3	0	27.51/0.7740	25.03/0.6674	24.60/0.6499	22.23/0.6450
RRDB-SFT + IKC [14]	×3	0	32.94/0.9104	29.14/0.8162	28.36/0.7814	26.34/0.8049
RRDB-SFT + MANet (ours)	×3	0	33.69/0.9184	29.81/0.8270	28.80/0.7931	27.39/0.8331
RRDB-SFT + GT (upper bound)	×3	0	34.12/0.9218	30.20/0.8338	28.98/0.7980	28.01/0.8463
HAN [30]	×4	0	21.71/0.5941	20.42/0.4937	21.48/0.4901	19.01/0.4676
DIP [38]	×4	0	26.71/0.7417	24.52/0.6360	24.34/0.6160	21.85/0.6155
KernelGAN [3]	×4	0	-	-	18.24/0.3689	16.80/0.3960
HAN [30] + Correction [20]	×4	0	24.31/0.6357	24.44/0.6341	24.01/0.6005	22.32/0.6368
RRDB-SFT + IKC [14]	×4	0	31.08/0.8781	27.83/0.7663	27.12/0.7233	25.16/0.7609
RRDB-SFT + MANet (ours)	×4	0	31.54/0.8876	28.28/0.7727	27.35/0.7305	25.66/0.7759
RRDB-SFT + GT (upper bound)	×4	0	31.93/0.8915	28.53/0.7786	27.48/0.7340	26.10/0.7872
HAN [30]	×4	15	20.88/0.4245	18.91/0.2901	21.01/0.4881	19.31/0.3552
DIP [38]	×4	15	18.60/0.2695	18.14/0.2392	17.90/0.2073	18.82/0.3476
KernelGAN [3]	×4	15	-	-	19.56/0.4582	13.65/0.1136
HAN [30] + Correction [20]	×4	15	19.21/0.2281	18.21/0.2478	19.25/0.4231	19.01/0.3500
RRDB-SFT + IKC [14]	×4	15	27.23/0.7877	25.55/0.6717	25.15/0.6236	23.31/0.6697
RRDB-SFT + MANet (ours)	×4	15	27.57/0.7918	25.75/0.6746	25.30/0.6259	23.56/0.6758
RRDB-SFT + GT (upper bound)	×4	15	27.81/0.7970	25.92/0.6787	25.38/0.6295	23.82/0.6861
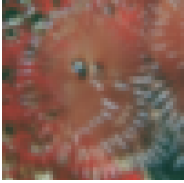
PSNR (dB)	20.86/21.28/21.94/-	25.19/23.92/25.34/-	17.28/19.99/17.22/-	25.75/24.91/26.24/-	26.71/25.84/27.42/-	27.03/26.18/27.77/-
LR (×4)	HAN [30]	DIP [38]	KernelGAN [3]	RRDB-SFT + IKC [14]	RRDB-SFT + **MANet** (ours)	RRDB-SFT + GT (upper bound)

Figure 6: Visual results of different methods on spatially variant blind SR and real-world images for scale factor 4. From the first to the third row, the corresponding kernel types are 2, 3 and 4, respectively.

images, it only has similar performance to HAN and DIP. As a learning-based method, IKC performs better, but it is inevitably affected by less discriminative patches because it predicts one kernel for the whole image. In comparison, the proposed MANet remedies the problem by estimating different kernels for different image patches, outperforming IKC by significant margins.

4.6. Experiments on Real-World SR

As there is no ground-truth for real images, we only compare visual results of different methods. Note that we only use L_1 pixel loss (no GAN loss) in training for simple and fair comparison. As shown in Fig. 6, similar to the results on synthetic images, HAN still generates blurry images. Different from HAN, DIP and KernelGAN produce images with obvious ringing artifacts. As for IKC, it over-sharpens the image and has obvious artifacts on edges, maybe due to the fact that it only estimates one kernel for different regions. In comparison, MANet produces sharp and natural edges with less artifacts based on the same non-blind SR model. The possible reason is that MANet estimates spatially variant kernels and feeds them to the non-blind model, which adaptively adds high-frequency details to edges and low-frequency information to flat areas. More visual results are provided in the supplementary.

5. Conclusion

In this paper, we proposed a mutual affine network (MANet) for spatially variant blind SR kernel estimation. MANet is composed of feature extraction and kernel reconstruction modules, and it has a moderate receptive field so as to keep the locality of degradation. In particular, it uses the proposed mutual affine convolution (MAConv) layer to exploit the channel interdependence by learned affine transformations between different channel splits, which can enhance model expressiveness without increasing the model receptive field, model size and computation burden. We conduct extensive experiments on synthetic datasets (including both spatially variant and invariant degradation) and real-world images to demonstrate the effectiveness of MANet. It performs well on blur kernel estimation, leading to state-of-the-art performance on blind image SR when MANet is combined with existing non-blind SR models. In the future, we will consider more real-world degradations and utilize GAN-based training for better visual quality.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[2] Yuval Bahat, Netalee Efrat, and Michal Irani. Non-uniform blind deblurring by reblurring. In *IEEE Conference on International Conference on Computer Vision*, pages 3286–3294, 2017.

[3] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*, pages 284–293, 2019.

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference*, pages 135.1–135.10, 2012.

[5] Wenlong Cheng, Mingbo Zhao, Zhiling Ye, and Shuhang Gu. Mfagan: A compression framework for memory-efficient on-device super-resolution gan. *arXiv preprint arXiv:2107.12679*, 2021.

[6] Victor Cornillere, Abdelaziz Djelouah, Wang Yifan, Olga Sorkine-Hornung, and Christopher Schroers. Blind image super-resolution with spatially variant degradations. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[7] Xin Deng, Yutong Zhang, Mai Xu, Shuhang Gu, and Yiping Duan. Deep coupled feedback network for joint exposure fusion and image super-resolution. *IEEE Transactions on Image Processing*, 30:3098–3112, 2021.

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199, 2014.

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

[10] Michael Elad and Arie Feuer. Restoration of a single super-resolution image from several blurred, noisy, and undersampled measured images. *IEEE Transactions on Image Processing*, 6(12):1646–1658, 1997.

[11] Sina Farsiu, Dirk Robinson, Michael Elad, and Peyman Milanfar. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2):47–57, 2004.

[12] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *IEEE Conference on International Conference on Computer Vision*, pages 349–356, 2009.

[13] Dong Gong, Wei Sun, Qinfeng Shi, Anton van den Hengel, and Yanning Zhang. Learning to zoom-in via learning to zoom-out: Real-world super-resolution by generating and adapting degradation. *arXiv preprint arXiv:2001.02381*, 2020.

[14] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1604–1613, 2019.

[15] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5407–5416, 2020.

[16] 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.

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

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

[20] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Correction filter for single image super-resolution: Robustifying off-the-shelf deep super-resolvers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1428–1437, 2020.

[21] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *European Conference on Computer Vision*, pages 645–660. Springer, 2020.

[22] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 466–467, 2020.

[23] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *IEEE Conference on International Conference on Computer Vision Workshops*, 2021.

[24] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *IEEE Conference on International Conference on Computer Vision*, 2021.

[25] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10601–10610, 2021.

[26] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *IEEE Conference on International Conference on Computer Vision Workshops*, pages 3408–3416. IEEE, 2019.

[27] Shunta Maeda. Unpaired image super-resolution using pseudo-supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 291–300, 2020.

[28] 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 *IEEE Conference on International Conference on Computer Vision*, pages 416–423, 2001.

[29] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *IEEE Conference on International Conference on Computer Vision*, pages 945–952, 2013.

[30] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020.

[31] Seobin Park, Jinsu Yoo, Donghyeon Cho, Jiwon Kim, and Tae Hyun Kim. Fast adaptation to super-resolution networks via meta-learning. *arXiv preprint arXiv:2001.02905*, 2020.

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

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[34] Wen-Ze Shao and Michael Elad. Simple, accurate, and robust nonparametric blind super-resolution. In *International Conference on Image and Graphics*, pages 333–348, 2015.

[35] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

[36] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3516–3525, 2020.

[37] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015.

[38] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[39] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10581–10590, 2021.

[40] Longguang Wang, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning parallax attention for stereo image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12250–12259, 2019.

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

[42] 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 Workshops*, pages 701–710, 2018.

[43] Yunxuan Wei, Shuhang Gu, Yawei Li, and Longcun Jin. Unsupervised real-world image super resolution via domain-distance aware training. *arXiv preprint arXiv:2004.01178*, 2020.

[44] Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 13385–13394, 2021.

[45] Yu-Syuan Xu, Shou-Yao Roy Tseng, Yu Tseng, Hsien-Kai Kuo, and Yi-Min Tsai. Unified dynamic convolutional network for super-resolution with variational degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12496–12505, 2020.

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

[47] 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 *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2018.

[48] 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, 2010.

[49] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3217–3226, 2020.

[50] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[51] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE Conference on International Conference on Computer Vision*, 2021.

[52] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[53] 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*, pages 3262–3271, 2018.

[54] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1671–1681, 2019.

[55] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 977–984, 2011.