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

Jingyun Liang
Guolei Sun
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 problem in computer vision. Recently, convolutional neural networks (CNNs) [6, 19, 37, 49, 27, 21, 45, 20] 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 [40, 3, 47, 12]. As a result, 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 [16] 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 [12, 3, 48, 47, 29, 22] 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 [34, 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-
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 [6, 46, 44]. However, most of existing networks have very large receptive fields, making them unsuitable for kernel estimation.

To tackle the problem, we propose the Mutual Affine Network (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 (MACConv) 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 MACConv 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., $22 \times 22$), it estimates kernels from tiny LR image patches. The minimum patch from which it can accurately estimate a kernel is of size $9 \times 9$.
- 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 [13, 31, 10, 50, 26]. 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. [22] 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. [5] propose SRVD 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. [47] propose a stretching strategy to take kernels as additional input and train an end-to-end model SRMD. Based on SRMD, Gu et al. [12] propose the SFTMD model that inputs kernels by the SFT layer [36], while Xu et al. [39] incorporate dynamic convolution into their UDVM model. Besides, Zhang et al. [42] 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. [32] 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 [41, 23, 11, 24, 18, 38], zero-shot SR [32, 33, 17] and data augmentation techniques [45]. 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 $I^{LR}$ is generated from the HR image $I^{HR}$ by a degradation model. When blur kernels are spatially invariant [8, 9], the relation between $I^{LR}$ and $I^{HR}$ is modelled as

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

(1)

where $\otimes$ denotes the convolution between $I^{HR}$ and blur kernel $k$, $\downarrow_s$ represents downsampling with scale factor $s$, and $n$ is noise. For blind SR, both HR image $I^{HR}$ and blur
kernel \( k \) are unknown. This problem is very ill-posed since many different pairs of \( I^{HR} \) and \( k \) can give rise to the same LR image \( I^{LR} \). 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

\[
I^{LR} = (K I^{HR}) \downarrow_s + n, \quad \text{(2)}
\]

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

In the literature of blind SR [3, 31, 12, 47, 40, 29, 13, 28, 32, 39, 45], kernels are widely assumed to be Gaussian as many estimated real-world kernels (e.g., [26, 31]) are actually unimodal and can typically be modeled by a Gaussian [29, 7, 40]. 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 [10, 50, 26], image patches blurred by different kernels have different patch distributions. KernelGAN [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 [30], feature extraction module is composed of convolution layers, residual blocks, a downsampler and an upsampler. The LR image is first input to a \( 3 \times 3 \) 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 non-linearity. 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 \( 3 \times 3 \) 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 \( K \in \mathbb{R}^{hw \times H \times W} \), where \( h, w, H, \text{and} \ W \) are kernel height, kernel width, HR image height and HR image width, respectively.

With elaborate architecture design, MANet has a moderate receptive field of \( 22 \times 22 \) 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 representational 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 (MACConv) layer to solve the problem.

Let \( \mathbf{x} \in \mathbb{R}^{C \times H \times W} \) be the input feature of the MACConv layer. As shown in Fig. 3, we first divide \( \mathbf{x} \) into \( S \) splits along the channel as

\[
\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_S = \text{split}(\mathbf{x}). \quad \text{(3)}
\]
For each split $x_i \in \mathcal{R}^{C_{in}(S-1) \times H_f \times W_f}$, we denote the concatenation of splits that are complementary to $x_i$ as $\bar{x}_i \in \mathcal{R}^{C_{in}(S-1) \times H_f \times W_f}$. Both $x_i$ and $\bar{x}_i$ are passed into the affine transformation module, which has a fully-connected network $\mathcal{F}$ to learn transformation parameters $\beta_i$ and $\gamma_i$ from $x_i$. Then, $\beta_i$ and $\gamma_i$ are used to scale and shift $x_i$, respectively. The whole process is formulated as

$$\beta_i, \gamma_i = \text{split}(\mathcal{F}(x_i)), \quad y_i = \beta_i \odot x_i + \gamma_i,$$

(4)

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

After transformation, for all $i \in \{1, 2, \ldots, S\}$, we use a $3 \times 3$ convolution layer to generate feature $z_i = \text{conv}_1(y_i)$. $z_i \in \mathcal{R}^{C_{out}(S-1) \times H_f \times W_f}$ when MACConv has $C_{out}$ output channels. Finally, different features $z_1, z_2, \ldots, z_S$ are concatenated to generate the MACConv output

$$z = \text{concat}(z_1, z_2, \ldots, z_S).$$

(5)

MACConv 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 $3 \times 3$ 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, MACConv only has $\frac{2}{S}C_{in}C_{out} + \frac{s^2-1}{2}C_{in}^2$ parameters and $(\frac{2}{S}C_{in}C_{out} + \frac{2(s^2-1)}{S}C_{in}^2)H_fW_f$ 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 MACConv is still the same as a single $3 \times 3$ convolution layer, as the affine transformation modules do not increase receptive field. In comparison, popular feature extraction blocks such as dense block [15] and squeeze-and-excitation (SE) block [14] 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

$$L = \frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{i=1}^{H} \sum_{j=1}^{W} \| K_{ij}^{(n)} - G_{ij}^{(n)} \|_1,$$

(6)

where $K_{ij}^{(n)}$ and $G_{ij}^{(n)}$ 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, 31, 12, 47, 40, 29, 13, 28, 32, 39, 45], we conduct experiments on $21 \times 21$ 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 $192 \times 192$ 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 [37] and SFT layers [36]. 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.
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.

### 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, accompanied 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 $22 \times 22$ to $38 \times 38$. 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.

### Table 1: Comparison of plain convolution, group convolution and MAConv.

| Type               | #Channel | #Split | LR Image PSNR/SSIM | #Params [M] | Memory [M] | FLOPs [G] | Runtime [s] |
|--------------------|----------|--------|--------------------|-------------|------------|-----------|-------------|
| Plain Convolution  | [32, 64, 32] | 2      | 46.42/0.9951       | 0.2557      | 234.229    | 12.6804   | 0.004699    |
|                    | [64, 128, 64] | -      | 47.65/0.9965       | 0.7649      | 248.170    | 33.9508   | 0.010534    |
|                    | [128, 256, 128] | 2     | 48.85/0.9974       | 2.5451      | 278.959    | 102.2613  | 0.018363    |
| Group Convolution  | [32, 64, 32] | 2      | 46.14/0.9910       | 0.2004      | 234.018    | 10.3683   | 0.004699    |
|                    | [64, 128, 64] | 2      | 46.72/0.9957       | 0.5437      | 247.326    | 26.7030   | 0.009985    |
|                    | [128, 256, 128] | 4     | 48.32/0.9969       | 1.6603      | 275.584    | 73.2701   | 0.012948    |
|                    | [128, 256, 128] | 6     | 47.96/0.9967       | 1.2160      | 273.896    | 58.7748   | 0.012562    |
| MAConv (ours)      | [32, 64, 32] | 2      | 45.87/0.9946       | 0.2102      | 234.068    | 11.7577   | 0.011263    |
|                    | [64, 128, 64] | 2      | 47.74/0.9965       | 0.5818      | 247.481    | 79.2125   | 0.016451    |
|                    | [128, 256, 128] | 4     | 49.39/0.9978       | 1.8104      | 276.162    | 78.1231   | 0.020956    |
|                    | [128, 256, 128] | 6     | 49.77/0.9979       | 1.5902      | 275.334    | 70.9173   | 0.025172    |

**Figure 4:** Comparison of different numbers of MACconv 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 $80 \times 80$. For all image patches, Gaussian kernel parameters $\sigma_1 = 10$ and $\sigma_2 = 0.7$. In particular, for different patches, $\theta$ is set to $\frac{\pi}{4}$ when the sum of spatial patch coordinates is even, and $\frac{3\pi}{4}$ otherwise.
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 $9 \times 9$. 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 [27], DIP [35], KernelGAN [3], HAN with correction [17], SRSVD [5], IKC [12] (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 $256 \times 256$ 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
its corresponding kernel is determined by a factor $\sigma = \frac{a}{m}$ and $\tau = \frac{b}{n}$, as shown in the table header. The best and second best results are highlighted in red and blue colors, respectively.

Table 3: Average PSNR/SSIM of different methods for spatially variant blind SR on BSD100 [25]. Every testing image is divided into $m \times n$ patches (patch size is 40 $\times$ 40), which are degraded by different kernels. According to experimental setup in Sec 4.1, for scale factor $s$, the Gaussian kernel width $\sigma$ and minimum kernel width $b$ are 2.325 and 0.175, respectively. In particular, for (patch $i, j$), its corresponding kernel is determined by $\sigma = \frac{a}{m}$ and $\tau = \frac{b}{n}$, 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 | 2 | 3 | 4 | 5 |
|-----------------|--------------|-------------|-----------------------------|---|---|---|---|---|
| HAN [27]        | $\times 2$   | 0           | $\sigma_1 = \frac{a + b}{\tau_0}$ | 24.98/0.624 | 25.31/0.6721 | 25.04/0.6593 | 25.41/0.6801 | 25.19/0.6643 |
| DIP [35]        | $\times 2$   | 0           | $\sigma_1 = \frac{a + b}{\tau_0}$ | 26.11/0.6765 | 25.04/0.6693 | 23.81/0.6375 | 25.05/0.6695 | 25.40/0.6780 |
| KernelGAN [3]   | $\times 2$   | 0           | $\sigma_1 = \frac{a + b}{\tau_0}$ | 24.81/0.6783 | 23.68/0.6391 | 21.88/0.5436 | 23.63/0.6410 | 23.49/0.6811 |
| HAN [27] + Correction [17] | $\times 2$   | 0           | $\sigma_1 = \frac{a + b}{\tau_0}$ | 27.88/0.7634 | 27.18/0.7252 | 25.53/0.6996 | 26.25/0.7101 | 25.80/0.6811 |
| SRSVD [5]       | $\times 2$   | 0           | $\sigma_1 = \frac{a + b}{\tau_0}$ | 28.53/0.8019 | 27.81/0.7871 | 24.70/0.7419 | 27.81/0.7765 | 27.81/0.7788 |
| RRDB-SFT + IKC [12] | $\times 2$   | 0           | $\sigma_1 = \frac{a + b}{\tau_0}$ | 28.45/0.7996 | 27.92/0.7922 | 27.49/0.7854 | 27.83/0.7946 | 27.75/0.7755 |
| RRDB-SFT + MANet (ours) | $\times 2$   | 0           | $\sigma_1 = \frac{a + b}{\tau_0}$ | 30.09/0.8397 | 30.70/0.8610 | 29.15/0.8205 | 30.46/0.8587 | 28.37/0.7975 |
| RRDB-SFT + GT (upper bound) | $\times 2$ | 0         | $\sigma_1 = \frac{a + b}{\tau_0}$ | 31.72/0.8570 | 31.57/0.8584 | 31.57/0.8584 | 28.37/0.7975 | 28.57/0.7625 |

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 [42] | BSD100 [25] | Urban100 [16] |
|-----------------|--------------|-------------|----------|------------|-------------|---------------|
| HAN [27]        | $\times 2$   | 0           | 26.83/0.7919 | 25.21/0.6888 | 25.11/0.6613 | 22.42/0.6571 |
| DIP [35]        | $\times 2$   | 0           | 28.19/0.7939 | 25.66/0.6999 | 25.03/0.6762 | 22.97/0.6737 |
| KernelGAN [3]   | $\times 2$   | 0           | 28.61/0.8013 | 23.92/0.6989 | 25.28/0.6955 | 21.97/0.6852 |
| HAN [27] + Correction [17] | $\times 2$   | 0           | 34.51/0.8787 | 31.10/0.8581 | 29.71/0.7993 | 28.08/0.7965 |
| SRSVD [5]       | $\times 2$   | 0           | 35.30/0.9381 | 31.48/0.8797 | 30.50/0.8545 | 28.62/0.8689 |
| RRDB-SFT + MANet (ours) | $\times 2$   | 0           | 35.95/0.9520 | 31.59/0.8845 | 31.95/0.8650 | 29.47/0.8777 |
| RRDB-SFT + GT (upper bound) | $\times 2$ | 0         | 36.52/0.9575 | 32.08/0.9064 | 31.40/0.8754 | 29.95/0.8677 |
| HAN [27]        | $\times 3$   | 15          | 20.58/0.1348 | 20.28/0.1078 | 20.53/0.1359 | 20.90/0.1286 | 20.34/0.1346 |
| DIP [35]        | $\times 3$   | 15          | 18.15/0.1854 | 18.14/0.2042 | 17.71/0.1960 | 18.02/0.1997 | 18.10/0.1998 |
| KernelGAN [3]   | $\times 3$   | 15          | 15.16/0.9992 | 14.60/0.9961 | 14.51/0.8173 | 15.11/0.8106 | 14.66/0.8089 |
| HAN [27] + Correction [17] | $\times 3$   | 15          | 18.13/0.1840 | 18.21/0.2141 | 18.04/0.2450 | 18.32/0.2209 | 18.41/0.2281 |
| RRDB-SFT + IKC [12] | $\times 3$   | 15          | 24.64/0.5950 | 24.94/0.6162 | 24.81/0.6017 | 25.01/0.6174 | 24.95/0.6078 |
| RRDB-SFT + MANet (ours) | $\times 3$   | 15          | 24.89/0.6030 | 25.21/0.6192 | 25.11/0.6167 | 25.24/0.6209 | 25.05/0.6118 |
| RRDB-SFT + GT (upper bound) | $\times 3$ | 15         | 25.86/0.6082 | 25.50/0.6255 | 25.53/0.6252 | 25.54/0.6264 | 25.50/0.6251 |

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.
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.

Acknowledgements We acknowledge Dr. Thomas Probst for insightful discussion. This work was partially supported by the ETH Zurich Fund (OK), a Huawei Technologies Oy (Finland) project, the China Scholarship Council and an Amazon AWS grant. Special thanks goes to Yijue Chen.
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] Victor Cornillere, Abdelaziz Djelouah, Wang Yifan, Olga Sorkine-Hornung, and Christopher Schroers. Blind image super-resolution with spatially variant distortations. ACM Transactions on Graphics (TOG), 38(6):1–13, 2019.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] Jiejun 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4681–4690, 2017.

[20] 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.

[21] 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.

[22] Jingyun Liang, Kai Zhang, Shuhan 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.

[23] 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.

[24] 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.

[25] 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.

[26] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In IEEE Conference on International Conference on Computer Vision, pages 945–952, 2013.
[27] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shu zhen 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.

[28] Seo bin Park, Jinsu Yoo, Donghyeon Cho, Ji won Kim, and Tae Hyun Kim. Fast adaptation to super-resolution networks via meta-learning. arXiv preprint arXiv:2001.02905, 2020.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] Jae Woong Soh, Sun woo 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Kai Zhang, Jinyoung 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.

[46] 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.

[47] 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.

[48] 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.

[49] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In European Conference on Computer Vision, pages 286–301, 2018.

[50] 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.