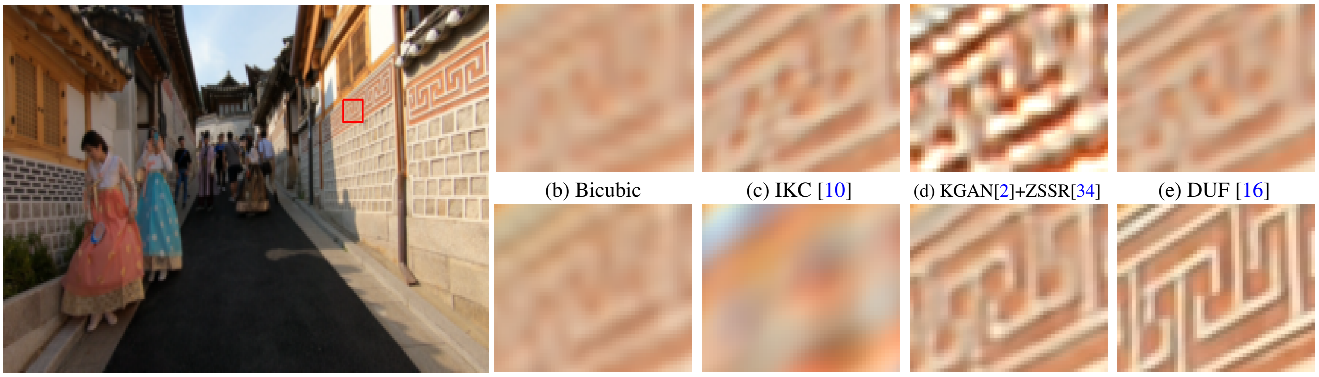
Deep Blind Video Super-resolution

Jinshan Pan[1] Haoran Bai[1] Jiangxin Dong[1] Jiawei Zhang[2] Jinhui Tang[1*]
[1]Nanjing University of Science and Technology [2]SenseTime Research



(a) Low-resolution (LR) input frame (b) Bicubic (c) IKC [10] (d) KGAN[2]+ZSSR[34] (e) DUF [16] (f) TOFlow [43] (g) Deblur[30]+EDVR[12] (h) w/o kernel modeling (i) Ours

Figure 1. Blind video super-resolution results (×4). Existing video super-resolution algorithms usually assume the blur kernel in the degradation is known or predefined and do not model the blur kernel in the restoration process. We show that blind image super-resolution methods do not handle the video super-resolution problem well (see (c)-(d)), while existing video super-resolution methods without modeling the blur kernel does not effectively capture the intrinsic characteristics of the video super-resolution problem which thus leads to over-smoothed results (see (e)-(h)). Our algorithm explicitly estimates blur kernels from low-resolution videos, which is able to generate clearer results with finer structural details.

Abstract

Existing video super-resolution (SR) algorithms usually assume that the blur kernels in the degradation process are known and do not model the blur kernels in the restoration. However, this assumption does not hold for blind video SR and usually leads to over-smoothed super-resolved frames. In this paper, we propose an effective blind video SR algorithm based on deep convolutional neural networks (CNNs). Our algorithm first estimates blur kernels from low-resolution (LR) input videos. Then, with the estimated blur kernels, we develop an effective image deconvolution method based on the image formation model of blind video SR to generate intermediate latent frames so that sharp image contents can be restored well. To effectively explore the information from adjacent frames, we estimate the motion fields from LR input videos, extract features from LR videos by a feature extraction network, and warp the extracted features from LR inputs based on the motion fields. Moreover, we develop an effective sharp feature exploration method which first extracts sharp features from restored intermediate latent frames and then uses a transformation operation based on the extracted sharp features and warped features from LR inputs to generate better features for HR video restoration. We formulate the proposed algorithm into an end-to-end trainable framework and show that it performs favorably against state-of-the-art methods.

1. Introduction

Blind video super-resolution (SR) aims to estimate high-resolution (HR) frames from a low-resolution (LR) sequence with unknown blur kernels. It is a fundamental problem in the vision and graphics communities and has received active research efforts within the last decade as high-definition devices have been widely used in our daily lives. As the HR sequences are usually contaminated by unknown blur, it is quite challenging to restore HR videos from low-resolution sequences.

Since blind video SR is an ill-posed problem, conventional methods usually develop kinds of hand-crafted priors to make this problem well-posed and estimate latent HR image in a variational approach [9, 1, 5, 33, 23, 27]. In spite of achieving decent results, these algorithms usually need to solve complex energy functions or involve complicated matching processes, and the performance is limited by the hand-crafted priors. In addition, most of these algorithms usually assume that the blur kernel is known or predefined (e.g., Bicubic kernel) and do not model blur kernels in the restoration, which cannot effectively capture the intrinsic

[*] Corresponding author.

characteristics of video SR [23].

Motivated by the first end-to-end trainable network for single image SR [7], lots of methods based on deep convolutional neural networks (CNNs) have been proposed [18, 8, 11, 46, 22, 20]. These approaches achieve decent results in single image SR, but cannot be easily applied to the video SR problem as the temporal information are not considered. To overcome this problem, most existing algorithms focus on developing effective motion fields and alignment estimation methods. For example, the subpixel motion compensation based on optical flow [37], deformable alignment networks [38, 40], and spatial alignment networks [24, 4, 43]. To better restore latent frames, the recurrent approaches and Generative Adversarial Networks (GANs) have been developed [6, 25]. These methods significantly promote the progress of video SR. However, they usually assume the blur kernel is known and fixed (e.g., Bicubic kernel), which does not model unknown blur kernels and thus leads to over-smoothed results when handling the blind video SR problem (Figure 1(e)-(f)). In addition, simply combining existing deblurring and video SR methods does not solve the blind SR problem well as shown in Figure 1(g).

Instead of assuming known blur kernels, several algorithms explicitly estimate blur kernels for SR [28, 10, 48, 2]. These algorithms show that using the estimated blur kernels for image SR is able to improve the results significantly [28, 2]. However, these algorithms are mainly developed for single image SR which cannot be extended to video SR directly as shown in Figure 1(c)-(d). The methods [23, 27] simultaneously estimate underlying motion fields and blur kernels for image restoration. However, the performance is limited by the hand-crafted image priors. Moreover, these hand-crafted image priors usually lead to complex optimization problems which are difficult to solve.

To overcome the above problems, we propose an effective video SR algorithm that simultaneously estimates underlying blur kernels, motion fields, and latent HR videos by deep CNN models so that our method can not only avoid the hand-crafted priors but also effectively estimate blur kernels and motion fields for better video restoration. The proposed algorithm explicitly estimates blur kernels from LR input videos and then develops an effective image deconvolution model based on the image formation of video SR to generate intermediate latent frames with sharp structural details. To explore sharper structural details of the restored intermediate latent images and information of the adjacent frames, we fuse the features extracted from LR videos based on motion field estimation and transform the sharp features of the intermediate latent images for better HR video restoration. By training the proposed algorithm in an end-to-end manner, it is able to generate clearer images with finer structural details (Figure 1). To the best of our knowledge, this is the first algorithm that develops deep CNNs based on a variational approach for blind video SR.

The main contributions are summarized as follows:

- We propose an effective blind video SR algorithm that simultaneously estimates blur kernels, motion fields, and latent images by deep CNN models.
- We develop an effective image deconvolution method based on the image formation of video SR to generate intermediate latent frames with sharp structural details.
- We develop a sharp feature exploration method to explore sharp features from the restored intermediate latent frames for HR video restoration.
- We formulate the proposed algorithm into an end-to-end trainable network and show that it performs favorably against state-of-the-art methods on both benchmark datasets and real-world videos.

2. Related Work

We briefly discuss methods most relevant to this work and put this work in proper context.

Variational approach. Since video SR is highly ill-posed, early approaches mainly focus on developing effective priors [9, 1, 5, 33] on the HR images to solve this problem. As these methods usually use known blur kernels to approximate the real ones which will lead to over-smoothed results. Several methods [23, 27] simultaneously estimate motion fields, blur kernels, and latent images in a maximum a posteriori (MAP) framework. In [23], Liu and Sun solve video SR by a Bayesian framework, where the motion fields, blur kernels, latent images, and noise levels are estimated simultaneously. Ma et al. [27] propose an effective Expectation Maximization (EM) framework to jointly solve video SR and blur estimation. Although promising results have been achieved, these algorithms require solving complex optimization problems. In addition, the performance is limited by the hand-crafted priors.

Deep learning approach. Motivated by the success of deep learning-based single image SR [7, 18, 8, 11, 46, 22, 20], several methods [14, 21, 17, 4, 24, 37, 43, 16, 12, 40, 38] explore the spatio-temporal information for video SR. Huang et al. [14] develop an effective bidirectional recurrent convolutional network to model the long-term contextual information. Some algorithms [21, 17] first estimate motion fields based on the hand-crafted priors and then use a deep CNN model to restore high-quality images. In [4], Caballero et al. develop an effective motion compensation method to explore the spatio-temporal information for video SR. Liu et al. [24] develop a temporal adaptive neural network and a spatial alignment network to better explore the temporal information. In [37], Tao et al. propose an effective subpixel motion compensation layer based on the estimated motion fields for video SR. Xue et al. [43] demonstrate the effect of optical flow on video image restoration and propose a unified video restoration framework to solve general video restoration problems. Instead of explicitly using optical flow for alignment, Jo et al. [16] dynamically estimate upsampling filters. In [12], Haris et al. extend the deep back-projection method [11] by a recurrent network. Wang et al. [40] improve the deformable convolution [38] and develop an effective temporal and spatial attention method to solve video restoration. This algorithm wins the champions in the NTIRE19 video restoration [29]. To better model the temporal information, several methods develop the temporal group attention [15] and temporal frame interpolation [42] for video SR.

To generate more realistic images, GANs have been used to solve both single image [20, 31, 3] and video [6, 25] SR problems. These algorithms generate decent results on video SR. However, these algorithms either explicitly or implicitly assume that the blur kernels are known and do not model the blur kernels for SR, which accordingly leads to over-smoothed results.

Estimating blur kernels has been demonstrated effective for image SR, especially for the details restoration [28, 10, 34, 2, 39, 45]. However, these algorithms are designed for single image SR. Few of them have been developed for video SR. Different from these methods, we propose a unified framework based on a deep CNN model to explicitly model blur kernels and explore the image formation of video SR to constrain the deep CNN model so that high-quality videos can be better-restored.

3. Revisiting Variational Methods

Instead of simply stacking deep neural networks to solve video SR, we develop an effective and compact deep CNN model by exploring the image formation of video SR for HR video restoration. To better motivate our algorithm, we first revisit how the variational methods [9, 23, 27] solve video SR and then introduce the proposed algorithm.

Following the definitions of [9, 23, 27], the degradation model for video SR is:

$$L_j = \mathbf{SKF}_{\mathrm{u}_{i\to j}} I_i + n_j, \tag{1}$$

where $\{L_j\}_{j=i-N}^{i+N}$ denote a set of LR images with $2N+1$ frames; I_i denotes the i-th HR frame; n_j denotes image noise, $\mathbf{S}$ and $\mathbf{K}$ denote the matrix form of the downsampling operation s and blur kernel K; $\mathbf{F}_{\mathrm{u}_{i\to j}}$ denotes the warping matrix w.r.t. optical flow $\mathrm{u}_{i\to j}$, and $\mathrm{u}_{i\to j}$ denotes the optical flow from I_i to I_j.

Based on the degradation model (1), the HR frame I_i, optical flow $\mathrm{u}_{i\to j}$, and blur kernel K can be estimated from $\{L_j\}_{j=i-N}^{i+N}$ by a Maximum a posteriori (MAP) [9, 23, 27]:

$$\begin{aligned}\{I_i^*, K^*, \{\mathrm{u}_{i\to j}^*\}\} &= \arg \max_{I_i, K, \{\mathrm{u}_{i\to j}\}} p(I_i, K, \{\mathrm{u}_{i\to j}\} | \{L_j\}),\\ &= \arg \max_{I_i, K, \{\mathrm{u}_{i\to j}\}} p(I_i) p(K) \prod_j p(\mathrm{u}_{i\to j})\\ & \quad p(L_i | I_i, K) \prod_{j\neq i} p(\{L_j\} | I_i, K, \{\mathrm{u}_{i\to j}\})\end{aligned} \tag{2}$$

By using hand-crafted image priors $\rho(I_i)$, $\varphi(\mathrm{u}_{i\to j})$, and $\phi(K)$ on the HR frame I_i, optical flow $\mathrm{u}_{i\to j}$, and blur kernel K, respectively, the video SR process can be achieved by alternatively minimizing [23]:

$$\begin{aligned} I_i^* = &\arg \min_{I_i} \|\mathbf{SK}I_i - L_i\| + \\ &\sum_{j=i-N, j\neq i}^{i+N} \|\mathbf{SKF}_{\mathrm{u}_{i\to j}} I_i - L_j\| + \rho(I_i),\end{aligned} \tag{3}$$

$$\mathrm{u}_{i\to j}^* = \arg \min_{\mathrm{u}_{i\to j}} \|\mathbf{SKF}_{\mathrm{u}_{i\to j}} I_i - L_j\| + \varphi(\mathrm{u}_{i\to j}), \tag{4}$$

and

$$K^* = \arg \min_K \|\mathbf{ST}_{I_i} K - L_i\| + \phi(K), \tag{5}$$

where $\mathbf{T}_{I_i}$ is a matrix of latent HR image I_i w.r.t. K [23].

Although the video SR algorithms [23, 27] based on above model have been demonstrated effective in both benchmark datasets and real-world videos, they need to define the hand-crafted image priors $\rho(I_i)$, $\varphi(\mathrm{u}_{i\to j})$, and $\phi(K)$ which usually lead to highly non-convex objective function (2). This makes the video SR problem more difficult to solve. In addition, the performance of video SR is limited by the hand-crafted image priors. We further note that most existing deep learning-based methods usually employ deep CNN models to solve video SR problem. Although these methods do not need to define hand-crafted priors, they cannot capture the intrinsic characteristics of video SR as the blur kernel is assumed to be known (e.g., Bicubic [40], Gaussian [32]). As the blur kernels in the degradation are complex [23], assuming known blur kernels usually leads to over-smoothed results.

To overcome these problems, we develop an effective deep CNN model which simultaneously estimates blur kernels, motion fields, and latent frame for video SR. The proposed model does not need the hand-crafted priors and can capture the intrinsic characteristics of the degradation process in video SR by modeling blur kernels. Thus, it can generate much better super-resolved videos with clearer structural details (Figure 1(i)).

4. Proposed Algorithm

The overview of the proposed method is shown in Figure 2. In the following, we explain the main ideas for each component in details. For simplicity, we use three frames to illustrate our method.

4.1. Optical flow estimation

The variational methods usually solve the problem (4) to generate optical flow and then use it to warp adjacent frames to the reference frame so that more reliable information can be used for the reference frame restoration. However, solving (4) needs to define the hand-crafted prior $\varphi(\mathrm{u}_{i\to j})$. In addition, the hand-crafted prior usually leads to a complex optimization problem which is difficult to solve. As optical

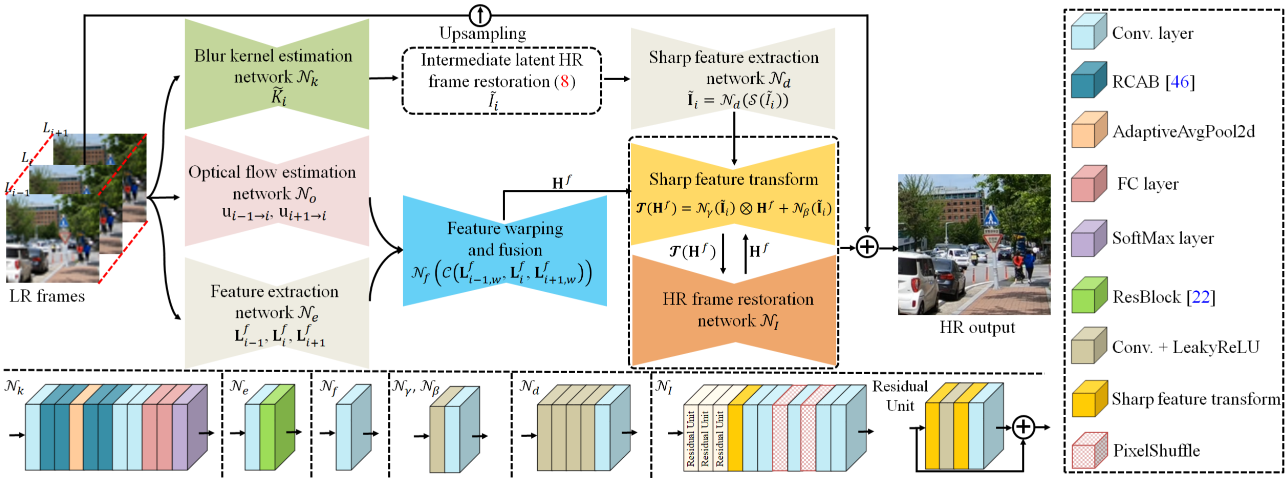
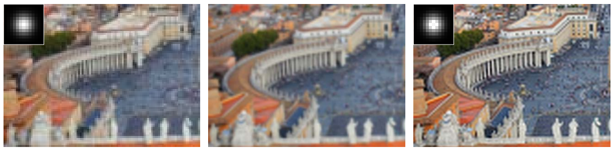

Upsampling
Blur kernel estimation network $\mathcal{N}_k$ $\tilde{K}_i$
Intermediate latent HR frame restoration (8) $\tilde{I}_i$
Sharp feature extraction network $\mathcal{N}_d$ $\tilde{\mathbf{I}}_i = \mathcal{N}_d(\mathcal{S}(\tilde{I}_i))$
L_{i+1}
L_i
L_{i-1}
LR frames
Optical flow estimation network $\mathcal{N}_o$ $\mathrm{u}_{i-1\to i}$, $\mathrm{u}_{i+1\to i}$
Feature extraction network $\mathcal{N}_e$ $\mathbf{L}^f_{i-1}, \mathbf{L}^f_i, \mathbf{L}^f_{i+1}$
$\mathbf{H}^f$
Feature warping and fusion $\mathcal{N}_f\left(\mathcal{C}(\mathbf{L}^f_{i-1,w}, \mathbf{L}^f_i, \mathbf{L}^f_{i+1,w})\right)$
Sharp feature transform $\mathcal{T}(\mathbf{H}^f) = \mathcal{N}_\gamma(\tilde{\mathbf{I}}_i) \otimes \mathbf{H}^f + \mathcal{N}_\beta(\tilde{\mathbf{I}}_i)$
$\mathcal{T}(\mathbf{H}^f)$
$\mathbf{H}^f$
HR frame restoration network $\mathcal{N}_I$
HR output
Conv. layer
RCAB [46]
AdaptiveAvgPool2d
FC layer
SoftMax layer
ResBlock [22]
Conv. + LeakyReLU
Sharp feature transform
PixelShuffle
$\mathcal{N}_k$
$\mathcal{N}_e$
$\mathcal{N}_f$
$\mathcal{N}_\gamma$, $\mathcal{N}_\beta$
$\mathcal{N}_d$
$\mathcal{N}_I$
Residual Unit
Residual Unit
Residual Unit
Residual Unit

Figure 2. An overview of the proposed method. It takes several adjacent frames as the input and super-resolves the center frame. First, we estimate frame-dependent blur kernels $\tilde{K}_i$ by $\mathcal{N}_k$ and generate the intermediate HR image ($\tilde{I}_i$) based on an image deconvolution method with $\tilde{K}_i$. In the meanwhile, we estimate optical flow between adjacent LR inputs and obtain the warped features ($\mathbf{L}^f_{i-1,w}$, $\mathbf{L}^f_{i+1,w}$) by applying the estimated optical flow to the extracted features from LR inputs (i.e., $\mathbf{L}^f_{i-1}$, $\mathbf{L}^f_{i+1}$). Then, we fuse the features $\mathbf{L}^f_{i-1,w}$, $\mathbf{L}^f_i$, and $\mathbf{L}^f_{i+1,w}$ to obtain $\mathbf{H}^f$, extract the sharp features from $\tilde{I}_i$ (i.e., $\mathcal{N}_d(\mathcal{S}(\tilde{I}_i))$), and generate $\mathcal{T}(\mathbf{H}^f)$ by a sharp feature transform based on $\mathbf{H}^f$ and $\mathcal{N}_d(\mathcal{S}(\tilde{I}_i))$ for HR frame restoration. Finally, we embed sharp feature transform into the restoration network $\mathcal{N}_I$ and restore the HR frame by adding the upsampled result of L_i to the output of $\mathcal{N}_I$. The proposed algorithm is jointly trained in an end-to-end manner. The mathematical operators are detailed in the main contents. For simplicity, we use three adjacent frames as an example.



(a) LR & GT kernel (b) Bicubic (c) Deblurred & kernel

Figure 3. Effect of the intermediate latent image restoration and blur kernel estimation. Using the estimated blur kernel to deblur LR images generates much sharper images (c).

flow can be efficiently estimated by deep neural networks, we use the PWC-Net [36] as the proposed optical flow estimation algorithm given its small model size and decent performance.

Given any three adjacent frames L_{i-1}, L_i, and L_{i+1}, the PWC-Net (denoted as $\mathcal{N}_o$ in Figure 2) is used to compute optical flow $\mathrm{u}_{i-1\to i}$ and $\mathrm{u}_{i+1\to i}$ based on the adjacent two input frames (L_{i-1}, L_i) and (L_{i+1}, L_i), where the PWC-Net for the computations of $\mathrm{u}_{i-1\to i}$ and $\mathrm{u}_{i+1\to i}$ shares the same parameters.

Based on the estimated optical flow, we perform the warping operation in a deep feature space instead of an image space. Let $\mathbf{L}^f_{i-1}$, $\mathbf{L}^f_i$, and $\mathbf{L}^f_{i+1}$ denote the features of L_{i-1}, L_i, and L_{i+1}, which are extracted by deep CNN model $\mathcal{N}_e$, we use the bilinear interpolation method to obtain the warped features $\mathbf{L}^f_{i-1}(\mathrm{x} + \mathrm{u}_{i-1\to i})$ and $\mathbf{L}^f_{i+1}(\mathrm{x} + \mathrm{u}_{i+1\to i})$ according to [36] (i.e., $\mathbf{L}^f_{i+1,w}$, $\mathbf{L}^f_{i-1,w}$ in Figure 2).

4.2. Blur kernel estimation

The blur kernel estimation can be achieved by solving (5). However, the accuracy of the blur kernels highly depends on whether the latent HR frames are accurate or not. In addition, using the hand-crafted prior $\phi(K)$ usually leads to a complex optimization problem which is difficult to solve. To overcome these problems, we develop a deep CNN model $\mathcal{N}_k$ to effectively estimate blur kernels.

Given the HR images $\{I_i\}$ and the corresponding LR images $\{L_i\}$, the proposed network $\mathcal{N}_k$ takes the LR images $\{L_i\}$ as input and outputs blur kernels. Different from [23] assuming that the blur kernels are all the same among different frames, we estimate frame-dependent blur kernels. To constrain the network $\mathcal{N}_k$, we develop a loss function based on (5):

$$\mathcal{L}_k = \|\mathbf{S}\tilde{\mathbf{K}}_i I_i - L_i\|_1, \tag{6}$$

where $\tilde{\mathbf{K}}_i$ denotes the matrix form of the output of the deep CNN model, i.e., $\mathcal{N}_k(L_i)$, and ℓ_1 norm is used. Figure 2 shows the network architectures of $\mathcal{N}_k$. The detailed parameters are included in the supplemental material.

The estimated blur kernels are shown in Figure 3(c), where the shape of the estimated one is visually close to that of the ground truth kernel. We will demonstrate the effectiveness of the blur kernel estimation in Section 6.

4.3. Latent frame restoration

With the blur kernel $\tilde{\mathbf{K}}_i$, we can estimate the HR frame from the input LR frame L_i according to (3). However, solving (3) needs to define the image prior. Instead of focusing on designing sophisticated image priors for HR frame restoration, we first restore an intermediate latent HR frame with sharp structural details and then develop a deep CNN model to explore these restored sharp structural details for better HR frame restoration. To this end, we first estimate the intermediate latent HR frame by:

$$\tilde{I}_i^* = \arg\min_{I_i} \|\mathbf{S}\tilde{\mathbf{K}}_i I_i - L_i\|^2 + \gamma \|\nabla I_i\|^2, \tag{7}$$

where $\|\nabla I_i\|^2$ is used to make the problem well-posed and L_2 norm is used so that it can be efficiently solved, and $\nabla = (\nabla_h, \nabla_v)^\top$ denotes the gradient operator, which includes horizontal and vertical operators.

Note that (7) is a least square problem. We can get the closed-form solution based on the fast Fourier transform (FFT) [44, 47] by:

$$\tilde{I}_i = \mathcal{F}^{-1}\left(\frac{1}{\gamma\triangle}\left(\mathrm{r} - \overline{\mathcal{F}(\tilde{K}_i)} \otimes_s \frac{\mathcal{P}(\frac{\mathcal{F}(\tilde{K}_i)\mathrm{r}}{\triangle})}{\mathcal{P}(\frac{\overline{\mathcal{F}(\tilde{K}_i)}\mathcal{F}(\tilde{K}_i)}{\triangle}) + \gamma}\right)\right), \tag{8}$$

where $\mathrm{r} = \overline{\mathcal{F}(\tilde{K}_i)}\mathcal{F}(L_i \uparrow^s)$; $\otimes_s$ and $\mathcal{P}$ denote the element-wise multiplication and the averaging operation at each $s \times s$ distinct block; $\uparrow^s$ denotes the s-folding upsampling operation; $\triangle = \overline{\mathcal{F}(\nabla_h)}\mathcal{F}(\nabla_h)) + \overline{\mathcal{F}(\nabla_v)}\mathcal{F}(\nabla_v)$.

Figure 3(c) shows the estimated intermediate HR image $\tilde{I}_i$. Note that though $\tilde{I}_i$ contains noise and artifacts, it also contains some clear contents which facilitate the following image restoration, especially for the structural details restoration (Figure 1(i)).

To better explore the sharp structural details of $\tilde{I}_i$ for final HR frame restoration, we develop a sharp feature exploration method based on a feature fusion module and a transformation operation by [41]. First, we fuse the warped features from LR frames by:

$$\mathbf{H}^f = \mathcal{N}_f(\mathcal{C}(\mathbf{L}^f_{i-1,w}, \mathbf{L}^f_i, \mathbf{L}^f_{i+1,w})), \tag{9}$$

where $\mathcal{C}$ denotes the concatenation operation, and $\mathcal{N}_f$ denotes a feature fusion network which contains one convolutional layer with filter size of 3×3 pixels and 128 feature channels. Then, we use the affine transformation [41] to generate features for HR video restoration by:

$$\mathcal{T}(\mathbf{H}^f) = \mathcal{N}_\gamma(\tilde{\mathbf{I}}_i) \otimes \mathbf{H}^f + \mathcal{N}_\beta(\tilde{\mathbf{I}}_i), \tag{10}$$

where $\tilde{\mathbf{I}}_i = \mathcal{N}_d(\mathcal{S}(\tilde{I}_i))$; $\mathcal{S}$ denotes the spatial-to-depth transformation which is used to ensure that $\tilde{\mathbf{I}}_i$ has the same spatial resolution as $\mathbf{H}^f$; $\otimes$ denotes the element-wise multiplication; $\mathcal{N}_\gamma$, $\mathcal{N}_\beta$, and $\mathcal{N}_d$ are the feature extraction networks. Finally, we develop a deep CNN model $\mathcal{N}_I$ with the residual unit [41] which takes (10) for the HR frame restoration. The network architectures of $\mathcal{N}_I$, $\mathcal{N}_\gamma$, $\mathcal{N}_\beta$, and $\mathcal{N}_d$ are shown in Figure 2.

4.4. Loss function

Given M video training pairs $\{\{L_i^m\}_{i=1}^Q, \{I_{gt,i}^m\}_{i=1}^Q\}_{m=1}^M$, where each video contains Q frames, we train the proposed network by minimizing:

$$\mathcal{L} = \sum_{m=1}^{M}\sum_{i=1}^{Q} \|\mathcal{N}(L_{i-N}^m; ...; L_i^m; ...; L_{i+N}^m) - I_{gt,i}^m\|_1 + \mathcal{L}_k, \tag{11}$$

where $\mathcal{N} = \{\mathcal{N}_o, \mathcal{N}_e, \mathcal{N}_f, \mathcal{N}_d, \mathcal{N}_\gamma, \mathcal{N}_\beta, \mathcal{N}_I\}$.

4.5. Implementation details

Training datasets. We train the proposed algorithm using the REDS dataset [29], where the REDS dataset contains 300 videos, each video contains 100 frames with an image size of 720×1280 pixels. Among 300 videos, 236 videos are used for training. Similar to the blind video SR settings [23], we first apply Gaussian kernels with the standard deviation to every frame of the original video and then downsample the filtered images by a factor of s according to the imaging process to generate LR blurry videos, where the standard deviation ranges from 0.4 to 2. During the training, we choose the first 50 consecutive frames from each video in the training dataset to train the proposed algorithm. We use the REDS4 dataset by [40] as our evaluation dataset, which does not overlap with the training dataset. In addition, we further use the Vid4 dataset [23] and SPMCS test dataset [37] as our test datasets to evaluate our model that is trained on the REDS dataset. We use the same way mentioned above to generate LR videos for test.

In addition to the commonly used Gaussian kernels that are widely adopted for evaluation in blind SR problems [10, 23, 26], we further evaluate our method using the blur kernels from [2] as its blur kernels are more realistic. To this end, we apply the generated blur kernels to the aforementioned dataset and generate the training and test datasets for evaluations. The blur kernels and videos in the test datasets do not overlap with those in the training datasets.

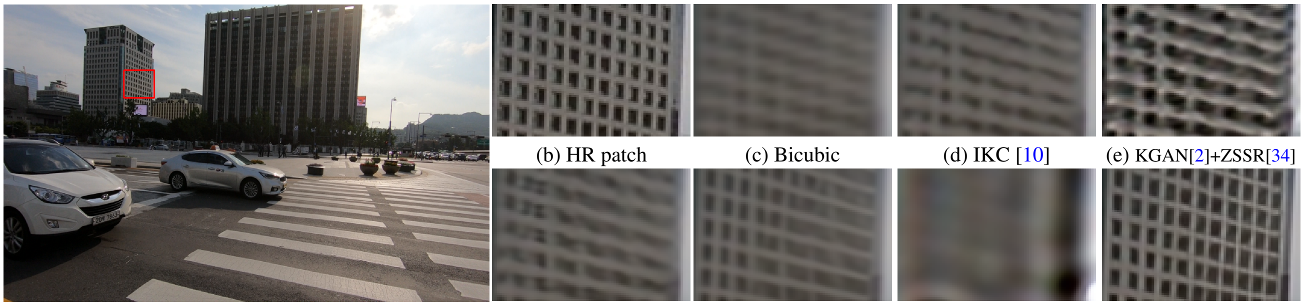
Parameter settings and training details. We empirically set $\gamma = 0.001$. The batch size is set to be 8. The size of each image patch is 64×64 pixels. In the training process, we use the ADAM optimizer [19] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The optical flow estimation network $\mathcal{N}_o$ is initialized by the pre-trained model [36]. We first train the blur kernel estimation network $\mathcal{N}_k$ from scratch and then jointly train the whole networks. The learning rates are initialized to be 10^{-4} except $\mathcal{N}_o$ and $\mathcal{N}_k$ because they are pretrained. We use 10^{-6} for $\mathcal{N}_o$ and $\mathcal{N}_k$. All the learning rates decrease to 0.5 times after every 100 epochs. We use the Bilinear upsampling operation to generate the upsampled result of the center frame. Similar to [41], we use the residual learning by taking $\mathcal{N}_\gamma(\tilde{\mathbf{I}}_i)$ as $\mathcal{N}_\gamma(\tilde{\mathbf{I}}_i) + 1$ when performing (10). The algorithm is implemented based on the PyTorch. More detailed network parameters and experimental results are included in the supplemental material. The source code and trained models are publicly available on the authors websites.

5. Experimental Results

In this section, we compare the proposed algorithm against state-of-the-art methods. Due to the page limit, we only show small portion results. More results are included in the supplemental material.

Table 1. Quantitative evaluations of the state-of-the-art video SR methods on the benchmark datasets with unknown Gaussian kernels.

Methods	Bicubic	RCAN [46]	SPMC [37]	DUF [16]	TOFlow [43]	RBPN [12]	EDVR [40]	Deblur [30]+EDVR [40]	KGAN [2]+ZSSR [34]	IKC [10]	MZSR [35]	Ours
REDS4 [40]	25.19 0.6898	27.15 0.7667	26.61 0.7485	27.19 0.7819	25.96 0.7171	27.15 0.7752	27.54 0.7898	17.93 0.4687	22.69 0.6601	27.36 0.7723	26.24 0.7274	**29.18** **0.8372**
Vid4 [23]	21.53 0.5551	23.36 0.6786	23.17 0.6726	23.91 0.7214	22.49 0.6183	23.45 0.6950	23.41 0.6902	15.58 0.3269	19.16 0.5519	23.52 0.6812	22.23 0.6004	**24.47** **0.7454**
SPMCS [37]	24.65 0.6740	27.10 0.7819	26.49 0.7593	27.07 0.7953	25.68 0.7221	26.86 0.7800	26.67 0.7739	17.81 0.4578	22.55 0.6658	27.03 0.7770	25.74 0.7161	**27.53** **0.8016**



(a) Ground truth HR image (b) HR patch (c) Bicubic (d) IKC [10] (e) KGAN[2]+ZSSR[34] (f) DUF [16] (g) RBPN [12] (h) Deblur[30]+EDVR[40] (i) Ours

Figure 4. Video SR result (×4) on the REDS dataset [29]. The proposed algorithm recovers high-quality frames with clearer structures.

Evaluations on the datasets with unknown Gaussian kernels. We compare the proposed algorithm against state-of-the-art methods including the deep CNN-based methods SPMC [37], DUF [16], TOFlow [43], RBPN [12], and EDVR [40]. As most existing deep learning-based video SR methods assume that the blur kernel is known and fixed Bicubic one, directly comparing with these methods may be unfair. Following the commonly used protocols (e.g., [10]), we further use our estimated blur kernels to generate deblurred frames according to the deconvolution method [30] and take the deblurred results as the input of these methods for fairness. In addition, we compare the proposed method with state-of-the-art deep CNNs-based single blind image SR methods including IKC [10], MZSR [35], and KGAN [2] with the ZSSR method [34]. We use the PSNR and SSIM as the evaluation metrics to evaluate the quality of each restored image on synthetic datasets. The PSNR and SSIM values of each restored image are calculated using RGB channels based on the script by [40].

Table 1 shows the quantitative evaluation results by the evaluated methods on the proposed benchmark datasets with unknown Gaussian kernels. Overall, our method outperforms the other algorithms by a large margin.

Figure 4 shows some results with a scale factor of 4 by the evaluated methods on the REDS4 dataset [29]. We note that the state-of-the-art video SR methods [16, 12] do not recover the correct structural details well as shown in Figure 4(f)-(g) because they do not model the blur kernel. In addition, the method that first uses the deblurring algorithm [30] to generate clear LR input with our estimated blur kernels and then restores HR videos by the deep video SR model does not generate clear frames (Figure 4(h)). This demonstrates that simply combining deblurring and video SR methods does not solve the blind video SR problems well. Although several methods [10, 2] explicitly estimate blur kernels to solve the blind SR problem, these methods are designed for single image and less effective for the video SR problem as shown in Figure 4(d)-(e). As the proposed algorithm develops a blur kernel estimation module which generates an intermediate latent HR image with sharp contents according to (7), thus facilitating clearer structural detail restoration as shown in Figure 4(i).

Evaluations on the datasets with blur kernels from [2]. We further evaluate our method using more realistic blur kernels, where the datasets are detailed in Section 4.5. Table 2 shows the evaluation results on the test dataset with blur kernels [2]. The state-of-the-art video SR methods [38, 15, 40] do not generate high-quality videos as they assume that the blur kernels in the degradation are the fixed Bicubic kernel. We note that the EDVR method [40] is effective on both the video SR problem and video deblurring problem. However, the video SR problem solved by [40] assumes that the blur kernel is the fixed Bicubic kernel while the video deblurring problem does not consider the downsampling degradation. Thus, it is less effective to super-resolve the LR images with unknown blur kernels. In addition, the commonly used baseline method that first applies the deblurring method to the LR videos and then uses non-blind video SR methods are not effective (see "Deblur [30]+EDVR [40]" in Table 2). In contrast, our method generates the results with higher PSNR and SSIM values, suggesting the effectiveness of the proposed method.

In addition, we evaluate our method with a larger model by using 3 ResBlocks and 20 Residual Units in $\mathcal{N}_e$ and $\mathcal{N}_I$. Table 2 shows that although using a larger model in the proposed method generates better results, the larger model usually involves more network parameters and needs more computational cost as shown in Table 7.

Evaluations of blur kernels. Different from existing video SR methods that use known blur kernels (e.g., fixed Bicubic [40] or Gaussian kernels [32]) in the video SR process, we develop a blur estimation method to estimate blur kernels for video SR. To examine the accuracy of the estimated

Table 2. Quantitative evaluations of the state-of-the-art video SR methods on the dataset ($\times 4$) with blur kernels from [2]. "Our-L" denotes that we use 3 ResBlocks and 20 Residual Units in $\mathcal{N}_e$ and $\mathcal{N}_I$.

Methods	Bicubic	RCAN [46]	SPMC [37]	DUF [16]	RBPN [12]	EDVR [40]	TGA [15]	TDAN [38]	Deblur [30]+EDVR [40]	KGAN [2]+ZSSR [34]	IKC [10]	DAN [26]	Ours	Ours-L
PSNR	25.68	27.46	27.18	27.32	27.62	28.26	27.70	27.48	19.55	26.35	27.52	27.68	29.36	**30.20**
SSIM	0.7147	0.7901	0.7842	0.8118	0.8125	0.8322	0.8199	0.7940	0.5436	0.7507	0.7864	0.7896	0.8495	**0.8702**

Table 3. Average kernel similarity of the estimated blur kernels.

Methods	KGAN [2]	Ours
Gaussian kernels	0.8772	**0.9983**
Blur kernels from [2]	0.7263	**0.9663**

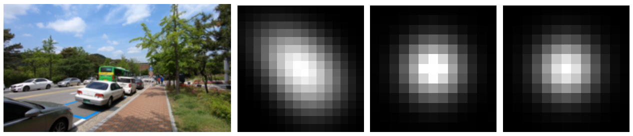

(a) LR frame (b) KGAN [2] (c) Ours (d) GT

Figure 5. Visualizations of the estimated blur kernels generated by the network $\mathcal{N}_k$ on the datasets with unknown Gaussian kernels.

Table 4. Effectiveness of the blur kernel estimation on video SR ($\times 4$). The results are obtained from the REDS4 test dataset.

Methods	Bilinear	$\tilde{I}_i$ by (7)	w/o kernel modeling	Bilinear of L_i as $\tilde{I}_i$	Ours
PSNR	24.73	26.65	28.87	29.04	**29.18**
SSIM	0.6712	0.7489	0.8280	0.8328	**0.8372**

blur kernels, we use the kernel similarity [13] as the metric and compare with the kernel estimation method [2] on the test dataset. Table 3 shows that the proposed method generates higher kernel similarity values than those by [2].

Figure 5 shows the visualizations of estimated blur kernels by $\mathcal{N}_k$. We note that the shape of the estimated blur kernel is visually similar to that of the ground truth kernel. Thus, both the quantitative and qualitative results demonstrate that the proposed algorithm is able to capture the degradation process well.

Real videos. We further qualitatively evaluate the proposed algorithm against state-of-the-art methods on real videos. Figure 6 shows a real example from [21]. The restored characters by state-of-the-art methods [10, 16, 12, 15, 38] are still blurry. In contrast, our algorithm generates the frame with clearer characters, which demonstrates that the proposed algorithm generalizes well.

6. Analysis and Discussions

We have shown that using the blur kernel estimation is able to help details restoration in video SR. In this section, we further analyze the effect of the proposed algorithm.

Effectiveness of the blur kernel estimation. As the most important part, the proposed blur kernel estimation process provides blur kernels, which thus leads to the intermediate latent images with clear contents for better details restoration. To demonstrate the effectiveness of this method, we disable this step in the proposed algorithm for fair comparisons. For this case, the baseline method (w/o kernel modeling) does not involve the blur kernel estimation and the intermediate latent HR frame restoration (7). Table 4 shows the quantitative evaluations on the benchmark datasets. The average PSNR of our method is 0.31dB higher than the method without blur kernel modeling on the REDS4 dataset, which demonstrates that using blur kernel estimation generates much better results.

Another question about the effect of the blur kernel estimation is that one may wonder whether the intermediate deblurred frame $\tilde{I}$ really helps the latent HR frame restoration. To answer this question, we replace our deblurred results with the commonly used Bilinear upsampled results of LR input frames [40] in our algorithm ("Bilinear of L_i as $\tilde{I}_i$ " in Table 4) for fair comparisons. Table 4 shows that using the Bilinear upsampled results of LR input frames does not generate good results, where the PSNR value of this baseline is 0.14dB lower than that of the proposed one.

In addition, we note that the quality of the intermediate deblurred frame (i.e., $\tilde{I}_i$ by (7)) is better than that of the Bilinear upsampled ones, where the PSNR value of the intermediate deblurred frames is 1.92dB higher than that of the Bilinear upsampled ones. This further indicates that using the blur kernel estimation is able to improve the performance of the super-resolved results.

The visualizations in Figure 7 further demonstrate that directly estimating the HR images using deep CNN models without modeling blur kernels does not generate the images with clearer structural details. In contrast, the proposed method generates much clearer images.

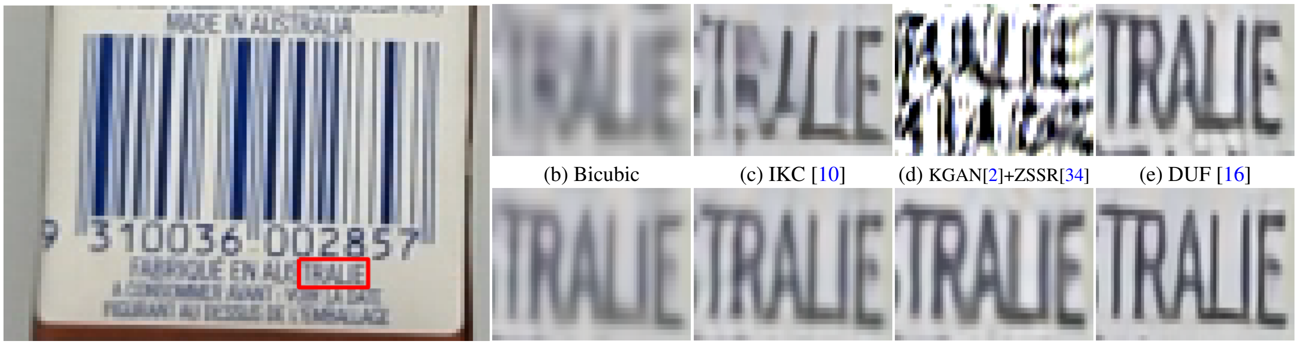
Effectiveness of the sharp feature exploration. We develop a sharp feature exploration method based on a feature fusion module and a sharp feature transform to estimate features for HR video restoration. One may wonder whether using a simple concatenation of the features from the LR frames and intermediate latent frames would generate the same or even better results. To answer this question, we further train an alternative method using the same experimental settings as our method, where the network $\mathcal{N}_I$ takes the concatenation of the features extracted from deblurred image by (8) and the warped features from LR input frames as the input. That is, the input of $\mathcal{N}_I$ is:

$$\mathcal{T}(\mathbf{H}^f) = \mathcal{N}_f\left(\mathcal{C}\left[\mathcal{N}_e\left(\mathcal{S}(\tilde{I}_i)\right); \mathbf{L}^f_{i-1,w}; \mathbf{L}^f_{i,w}; \mathbf{L}^f_{i+1,w}\right]\right). \tag{12}$$

Table 5 shows that using (12) does not generate good video super-resolution results compared to the proposed method.

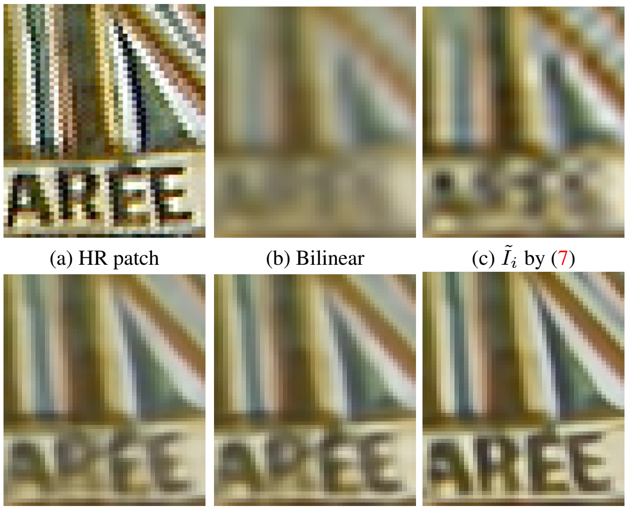
In addition, as we only use the deblurred result of the center frame ($\tilde{I}_i$) for HR frame restoration in (10), one may wonder whether using all the deblurred frames $\{\tilde{I}_i\}$ would improve the performance or not. To answer this question, we further evaluate the proposed method when all the deblurred frames $\{\tilde{I}_i\}$ are used for HR frame restoration. Table 5 shows that using all the deblurred frames $\{\tilde{I}_i\}$ does not improve the performance.

Relations with model-based methods. Several methods, e.g., [26, 44, 45], unfold the MAP-based variational model



(a) Input (b) Bicubic (c) IKC [10] (d) KGAN[2]+ZSSR[34] (e) DUF [16] (f) RBPN [12] (g) TDAN [38] (h) TGA [15] (i) Ours

Figure 6. Results (×4) on a real video with unknown blur. The proposed algorithm generates the frame with clearer characters.



(a) HR patch (b) Bilinear (c) $\tilde{I}_i$ by (7) (d) w/o kernel modeling (e) Bilinear of L_i as $\tilde{I}_i$ (f) Ours

Figure 7. Effectiveness of the blur kernel estimation on video SR (×4). Using blur kernel estimation is able to generate the results with clearer structural details.

Table 5. Effectiveness of the sharp feature exploration on video SR (×4). The results are obtained from the REDS4 dataset [40].

Methods	$\mathcal{T}(\mathbf{H}^f)$ by (12)	$\mathcal{T}(\mathbf{H}^f)$ w/ all $\{\tilde{I}_i\}$	$\mathcal{T}(\mathbf{H}^f)$ w/ only $\tilde{I}_i$ (Ours)
PSNR	28.98	29.13	**29.18**
SSIM	0.8325	0.8351	**0.8372**

into a deep CNN regularized model and an image reconstruction model for image SR, where these two models are alternatively solved to generate HR images. Different from these methods, we develop a deep CNN which takes the variational model as a differentiable module to solve blind video SR. The variational model (7) is used to generate sharp image contents which are further utilized by the proposed sharp feature exploration module for better HR frame restoration. Our network directly (instead of alternatively) estimates latent HR frames. In addition, these methods are designed for image SR and are less effective for the blind video SR task as demonstrated in Section 5.

Warping operation in the feature v.s. image spaces. We evaluate the effect of the warping operation in the feature space on the REDS4 test dataset. Table 6 shows that using the warping operation in the deep feature space (i.e., "Feature space warping") generates the results with higher PSNR and SSIM values.

Table 6. Effectiveness of the feature warping on video SR (×4). The results are obtained from the REDS4 dataset [40].

Methods	Image space warping	Feature space warping (Ours)
PSNR	28.98	**29.18**
SSIM	0.8312	**0.8372**

Table 7. Comparisons of model size and FLOPS. The results are tested on the images with 720 × 1280 pixels.

Methods	RCAN [46]	RBPN [12]	EDVR [40]	Ours-L	Ours
Model parameters (M)	15.59	**12.77**	20.63	20.54	14.08
FLOPs (G)	919.21	1245.42	1480.57	857.27	**485.34**

Model size and computational complexity. As our network design is motivated by the variational model-based methods instead of simply stacking deep neural networks for video SR, it has relatively fewer model parameters and the lowest floating point operations (FLOPs) as shown in Table 7. All these demonstrate that the performance gains are not due to the use of large capacity models.

7. Concluding Remarks

We have proposed an effective blind video SR algorithm which simultaneously estimates underlying blur kernels, motion fields, and latent HR videos by deep CNN models. The blur kernel estimation is able to estimate blur kernels from LR input videos. With the estimated blur kernels, we develop an effective image deconvolution method based on the image formation of video SR to generate intermediate latent HR frames with sharp structural details. We have proposed a sharp feature exploration method by exploring features from intermediate latent HR frames and input frames for better HR video restoration. We have shown that the proposed algorithm can be trained in an end-to-end manner and performs favorably against state-of-the-art methods.

Acknowledgements. This work has been supported in part by the National Key R&D Program of China (No. 2018AAA0102001), the National Natural Science Foundation of China (Nos. 61922043, 61872421, 61732007), the Natural Science Foundation of Jiangsu Province (No. BK20180471), and the Fundamental Research Funds for the Central Universities (No. 30920041109).

References

[1] Benedicte Bascle, Andrew Blake, and Andrew Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, pages 573–582, 1996. 1, 2

[2] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *NeurIPS*, pages 284–293, 2019. 1, 2, 3, 5, 6, 7, 8

[3] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a GAN to learn how to do image degradation first. In *ECCV*, pages 187–202, 2018. 3

[4] Jose Caballero, Christian Ledig, Andrew P. Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*, pages 2848–2857, 2017. 2

[5] Stephen C. Cain, Majeed M. Hayat, and Ernest E. Armstrong. Projection-based image registration in the presence of fixed-pattern noise. *IEEE TIP*, 10(12):1860–1872, 2001. 1, 2

[6] Mengyu Chu, You Xie, Laura Leal-Taixé, and Nils Thuerey. Temporally coherent gans for video super-resolution (tecogan). *CoRR*, abs/1811.09393, 2018. 2, 3

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2016. 2

[8] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pages 391–407, 2016. 2

[9] Sina Farsiu, M. Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE TIP*, 13(10):1327–1344, 2004. 1, 2, 3

[10] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *CVPR*, pages 1604–1613, 2019. 1, 2, 3, 5, 6, 7, 8

[11] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, pages 1664–1673, 2018. 2, 3

[12] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, pages 3897–3906, 2019. 1, 2, 3, 6, 7, 8

[13] Zhe Hu and Ming-Hsuan Yang. Learning good regions to deblur images. *IJCV*, 115(3):345–362, 2015. 7

[14] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *NeurIPS*, pages 235–243, 2015. 2

[15] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory G. Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *CVPR*, pages 8005–8014, 2020. 3, 6, 7, 8

[16] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, pages 3224–3232, 2018. 1, 2, 6, 7, 8

[17] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K. Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE TCI*, 2(2):109–122, 2016. 2

[18] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 2

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[20] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 105–114, 2017. 2, 3

[21] Renjie Liao, Xin Tao, Ruiyu Li, Ziyang Ma, and Jiaya Jia. Video super-resolution via deep draft-ensemble learning. In *ICCV*, pages 531–539, 2015. 2, 7

[22] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, pages 1132–1140, 2017. 2

[23] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE TPAMI*, 36(2):346–360, 2014. 1, 2, 3, 4, 5, 6

[24] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas S. Huang. Robust video super-resolution with learned temporal dynamics. In *ICCV*, pages 2526–2534, 2017. 2

[25] Alice Lucas, Santiago Lopez Tapia, Rafael Molina, and Aggelos K. Katsaggelos. Generative adversarial networks and perceptual losses for video super-resolution. *IEEE TIP*, 28(7):3312–3327, 2019. 2, 3

[26] Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. In *NeurIPS*, 2020. 5, 7

[27] Ziyang Ma, Renjie Liao, Xin Tao, Li Xu, Jiaya Jia, and Enhua Wu. Handling motion blur in multi-frame super-resolution. In *CVPR*, pages 5224–5232, 2015. 1, 2, 3

[28] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *ICCV*, pages 945–952, 2013. 2, 3

[29] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPR Workshops*, pages 1996–2005, 2019. 3, 5, 6

[30] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Deblurring images via dark channel prior. *IEEE TPAMI*, 40(10):2315–2328, 2018. 1, 6, 7

[31] Mehdi S. M. Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, pages 4491–4500, 2017. 3

[32] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *CVPR*, pages 6626–6634, 2018. 3, 6

[33] Oded Shahar, Alon Faktor, and Michal Irani. Space-time super-resolution from a single video. In *CVPR*, pages 3353–3360, 2011. 1, 2

[34] Assaf Shocher, Nadav Cohen, and Michal Irani. “zero-shot” super-resolution using deep internal learning. In *CVPR*, pages 3118–3126, 2018. 1, 3, 6, 7, 8

[35] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *CVPR*, pages 3513–3522, 2020. 6

[36] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 4, 5

[37] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, pages 4482–4490, 2017. 2, 5, 6, 7

[38] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. T-DAN: temporally-deformable alignment network for video super-resolution. In *CVPR*, pages 3357–3366, 2020. 2, 3, 6, 7, 8

[39] Cornillére Victor, Djelouah Abdelaziz, Yifan Wang, Sorkine-Hornung Olga, and Schroers Christopher. Blind image super-resolution with spatially variant degradations. In *SIGGRAPH Asia*, 2019. 3

[40] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *CVPR Workshops*, 2019. 2, 3, 5, 6, 7, 8

[41] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, pages 606–615, 2018. 5

[42] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *CVPR*, pages 3367–3376, 2020. 3

[43] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 127(8):1106–1125, 2019. 1, 2, 6

[44] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *CVPR*, pages 3214–3223, 2020. 5, 7

[45] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *CVPR*, pages 1671–1681, 2019. 3, 7

[46] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 294–310, 2018. 2, 6, 7, 8

[47] Ningning Zhao, Qi Wei, Adrian Basarab, Nicolas Dobigeon, Denis Kouame, and Jean-Yves Tourneret. Fast single image super-resolution using a new analytical solution for l2-l2 problems. *IEEE TIP*, 25(8):3683–3697, 2016. 5

[48] Ruofan Zhou and Sabine Süsstrunk. Kernel modeling super-resolution on real low-resolution images. In *ICCV*, pages 2433–2443, 2019. 2