Self-Supervised Deep Blind Video Super-Resolution

Haoran Bai and Jinshan Pan, Senior Member, IEEE

Abstract—Existing deep learning-based video super-resolution (SR) methods usually depend on the supervised learning approach, where the training data is usually generated by the blurring operation with known or predefined kernels (e.g., Bicubic kernel) followed by a decimation operation. However, this does not hold for real applications as the degradation process is complex and cannot be approximated by these idea cases well. Moreover, obtaining high-resolution (HR) videos and the corresponding low-resolution (LR) ones in real-world scenarios is difficult. To overcome these problems, we propose a self-supervised learning method to solve the blind video SR problem, which simultaneously estimates blur kernels and HR videos from the LR videos. As directly using LR videos as supervision usually leads to trivial solutions, we develop a simple and effective method to generate auxiliary paired data from original LR videos according to the image formation of video SR, so that the networks can be better constrained by the generated paired data for both blur kernel estimation and latent HR video restoration. In addition, we introduce an optical flow estimation module to exploit the information from adjacent frames for HR video restoration. Experiments show that our method performs favorably against state-of-the-art ones on benchmarks and real-world videos.

Index Terms—Self-supervised learning, blind video super-resolution, convolutional neural network, deep learning.

I. INTRODUCTION

With the rapid development of high definition devices, visualizing the videos generated by some low-resolution (LR) imaging devices on these high definition devices usually leads to significant aliasing and blur effect. Thus, it is of great interest to develop an effective algorithm to super-resolve videos for better visualization on the high definition devices.

The goal of video super-resolution (SR) is to infer the latent high-resolution (HR) videos from given LR ones. The degradation process of the video SR problem is usually modeled as [6]:

$$y_j = SK_j F_{i,j} x_i + n, j = i - N, i - N + 1, \ldots, i + N,$$

(1)

where \( y_j \), \( x_i \), and \( n \) denote the \( j \)-th LR frame, \( i \)-th HR frame, and noise, respectively; \( S \) and \( K_j \) denote the downsampling and blurring matrix w.r.t. \( S \) and \( K_j \); \( S \) and \( K_j \) denote the scale factor and blur kernel; \( F_{i,j} \) denotes the warping matrix which warps \( x_i \) to the \( j \)-th frame; \( N \) denotes the half-length of the video sequence. The degradation model (1) is a widely used classical model for blind video SR [6], [7], [8], which takes into account the blurring influenced by the sensor optics as well as by tiny camera motion, so each frame has a different blur kernel.

Video SR is a highly ill-posed problem as the latent HR frame, blurring matrix, and warping matrix are unknown.

Some recent significant advance has been achieved due to the use of kinds of deep convolution neural networks (CNNs). However, the ground truth HR videos are always required to train deep models, which cannot be easily satisfied in real applications. To constrain the deep model training, most video SR methods [5], [8], [9], [10], [11], [12], [13], [14], [15] assume that the blur kernels are known or predefined (e.g., Bicubic kernel) for synthesizing the paired datasets. Although they can achieve state-of-the-art results on existing benchmark datasets, as shown in Fig. 1(g) and (h), the deep models trained on the synthetic datasets generated by predefined blur kernels cannot be generalized well on real videos because blur kernels in real applications are much more complex. More visual results on real scenarios will be shown in Section IV.

To generate realistic blur kernels for the SR problem, several methods develop effective deep models to estimate blur kernels from input LR images [1], [3]. Another kind of methods develop effective unpaired learning [16] and zero-shot learning methods [2] to solve the SR problem. These methods achieve decent performance on real-world applications. However, most of these methods are designed for the single image SR problem, and few are developed for the video SR problem. Directly applying them to the video SR problem can not produce convincing results as shown in Fig. 1(d)–(f).

In this paper, we present an effective video SR algorithm based on self-supervised learning, which can simultaneously estimate the blur kernels and the latent HR frames based on deep CNNs. Instead of synthesizing unpaired datasets [16] which usually requires sophisticated network designs, our algorithm only explores the information from the LR input videos for HR ones restoration. We first develop deep CNN models to estimate blur kernels and latent HR videos from LR input videos. Then, we use the original LR input videos to constrain the regenerated LR videos, which are generated by the estimated blur kernels and latent HR videos according to the image formation of video SR, so that the deep models can be learned. However, directly using LR input videos as the supervision of network
training usually leads to trivial solutions. To address this issue, we first explore the sparse property of blur kernels to constrain the blur kernel estimation network, and then develop a simple and effective method for generating auxiliary paired data from the original LR input videos based on the image formation of video SR, so that the networks can be better constrained by both the generated paired data and sparse property constraint for both blur kernel estimation and latent HR video restoration. In addition, we introduce an optical flow estimation module to exploit the information from adjacent frames for better HR video restoration. By training the proposed algorithm in an end-to-end fashion, we show that it performs favorably against state-of-the-art methods on benchmark datasets and real-world videos. To the best of our knowledge, this is the first algorithm that develops a self-supervised learning method for blind video SR. Fig. 1 shows one super-resolved frame, where the proposed method generates the results with correct structural details than both single and video SR methods.

The main contributions are summarized as follows:

- We propose an effective self-supervised learning algorithm for video SR that does not require any paired or unpaired datasets as the supervision.
- To constrain the deep models for video SR, we develop a simple and effective method to generate auxiliary paired data from the original LR input videos according to the image formation of video SR.
- We train our method in an end-to-end manner and show that it generates favorable results on both benchmark datasets and real-world videos. To the best of our knowledge, this is the first self-supervised learning-based algorithm for blind video SR.

**II. RELATED WORK**

A. **Video SR Based on Known Blur Kernels**

Due to the ill-posed nature of the video SR problem, developing kinds of priors or regularization has been the focus of much in the past decade, and significant progress has been made [17], [18], [19], [20]. However, using priors usually leads to complex optimization problems which are difficult to solve. Motivated by the success of the deep CNNs in single image SR [21], [22], [23], recent methods usually use deep CNNs to solve the video SR problem with motion compensation [8], [9], [10], [11], [12], [13]. For example, Caballero et al. [12] propose an effective up-sampling and motion compensation method based on the single image SR [21] for real-time video SR. Tao et al. [8] propose an effective sub-pixel motion compensation layer based on the image formation model of the video SR, which is able to restore structural details. Instead of using explicit motion compensation, Jo et al. [14] learn dynamic upsampling filters and a residual image for effective to restore HR videos. To better explore the useful information from adjacent frames, the temporal group attention [24] and deformable alignment network [5], [15] have been proposed. Although these aforementioned methods achieve decent performance, they usually assume that the blur kernels are known or predefined (e.g., Bicubic kernels). However, the blur kernels for real images are much more complicated. Thus, these methods cannot be directly applied to real-world applications.

B. **Blind Video SR**

Instead of assuming the blur kernels are known, several methods estimate blur kernels from given LR videos. In [25], Tomer and Michael estimate blur kernels by relying on patch recurrence across scales of the input LR images under a MAP framework. In [6], Liu and Sun develop an effective Bayesian adaptive video SR algorithm, where the blur kernels are directly estimated from given LR videos in a Bayesian framework. Ma et al. [7] estimate blur kernels to super-resolve the blurry LR videos. Although decent performance has been achieved, these methods usually need to solve complex optimization problems, and the performance is limited to the hand-crafted priors.

Instead of using hand-crafted priors, several methods develop deep CNNs to estimate blur kernels for single image SR [1], [3]. These methods have been shown better results than the hand-crafted prior-based methods [6], [7], [25]. However, they are designed for single image SR which cannot be applied to video SR. In [26], Pan et al. propose a blind video SR method that simultaneously estimates blur kernels and latent HR frames and develop an image deconvolution method for generating sharp intermediate frames to guide the latent HR frames restoration. Although this method can achieve decent results, it still depends...
on the supervised learning approach and requires the ground truth HR videos for supervision, which cannot be easily satisfied in real applications.

In this work, we aim to solve blind video SR in real-world applications lacking paired training datasets, and how to effectively train deep models is the key challenge in this task. Different from these aforementioned methods which approximate real-world scenarios by using synthetic paired data for supervision or develop variational models based on hand-crafted priors, we propose a simple and effective self-supervised learning approach for blind video SR, which is able to simultaneously estimate blur kernels and HR videos from input LR videos without any paired or unpaired training datasets.

C. Self-Supervised Learning-Based Methods

Self-supervised learning has been widely developed to solve the image restoration problem (e.g., image denoising [27]) when the paired training data is not available. In the SR problem, Bulat et al. [28] first use a GAN to synthesize paired training datasets and then use the paired training datasets as the supervision for image SR. Maeda [16] develops an effective unpaired image SR algorithm based on a pseudo-supervision in a unified framework. These methods perform well on real-world applications. However, they are developed for single image SR, which cannot be directly extended to the video SR problem. In [29], Chen et al. compute blur kernels from the estimated optical flow and propose a self-supervised loss for video deblurring. However, it still requires an additional paired dataset with ground-truth sharp images for supervision.

In contrast, the proposed method enables fully self-supervised learning for blind video SR, where the blur kernels and the HR videos are estimated simultaneously so that the HR videos can be better restored.

III. PROPOSED ALGORITHM

Given the LR sequence \( \{y_i\} \), the proposed method aims to estimate the HR sequence \( \{x_i\} \) without any supervision of the ground truth HR sequences. For simplicity, we assume that the latent HR frame \( x_i \) is estimated by \( \{y_{i-N}, \ldots, y_{i-1}, y_i, y_{i+1}, \ldots, y_{i+N}\} \), where \( N \) denotes half length of the input LR sequence. Based on the image formation model (1), recovering the HR frame \( x_i \) needs to estimate the blur kernel and warping matrix (w.r.t. optical flow). Therefore, we develop an effective self-supervised learning approach so that the blur kernels, the optical flow, and the latent HR frames can be simultaneously estimated without any HR sequence supervision. Based on the proposed self-supervised learning approach, the deep CNN model is designed as two branches. The main branch is used to estimate blur kernel, the optical flow, and latent HR frame, and the auxiliary branch uses the auxiliary paired training data which is generated based on the LR input frames and estimated blur kernel from the main branch to constrain the network training for the optical flow and latent HR frame. All the branches are jointly trained in an end-to-end manner based on the self-supervised learning method. Fig. 2 shows an overview of the proposed algorithm. In the following, we first introduce the network designs about the blur kernel estimation, optical flow estimation, and latent HR frame restoration and then present the proposed self-supervised learning approach to solve the blind video SR problem.

A. Blur Kernel Estimation

The blur kernel estimation module aims to estimate the blur kernel \( K_i \) from \( \{y_{i-N}, \ldots, y_{i-1}, y_i, y_{i+1}, \ldots, y_{i+N}\} \). Let \( \mathcal{N}_k \) denotes the blur kernel estimation network, and we estimate blur kernel \( K_i \) by:

\[
K_i = \mathcal{N}_k \left( C \left[ y_{i-N}, \ldots, y_{i-1}, y_i, y_{i+1}, \ldots, y_{i+N} \right] \right),
\]

where \( C[\cdot] \) denotes a concatenation operation. For the network \( \mathcal{N}_k \), we first adopt a convolutional network with a pooling operation to extract the information of the blur kernel from the input. In this convolutional network, we adopt the residual channel attention block (RCAB) proposed by [23], which is widely used for restoration tasks and has been shown to be effective for blur kernel estimation in [30]. Then, we apply two fully connected layers followed by a softmax activation function to obtain the blur kernel and ensure that the sum of all elements is 1. The detailed network configurations are shown in Fig. 3.

B. Optical Flow Estimation

The optical flow estimation is mainly used to compute the warping matrix so that the information of the adjacent frames can be used for better latent HR frame restoration. In this work, we use the PWC-Net [31] as our optical flow estimation model because it is effective in some video restoration tasks (e.g., video deblurring [32]). As the latent HR sequence \( \{x_i\} \) is not available, we compute the optical flow from the LR input sequence by:

\[
u_{j\rightarrow i} = \mathcal{N}_f \left( y_j, y_i \right), \quad j = i - N, \ldots, i - 1, i + 1, \ldots, i + N,
\]

where \( \mathcal{N}_f \) denotes the optical flow estimation network, which adopts the default network configurations of [31]. With the estimated optical flow \( u_{j\rightarrow i} \), the warping operation, i.e., \( F_{j\rightarrow i}(y_j) \), can be computed by applying the bilinear interpolation to \( y_j \). Similar to the warping operation in [31], we perform the warping operation on the features of LR input frames, where the features are extracted by the feature extraction network \( \mathcal{N}_e \) as shown in Fig. 2. The detailed network configurations of \( \mathcal{N}_e \) are shown in Fig. 3.

C. Latent HR Frame Restoration

Given the warped features of LR input frames, we estimate the latent frame \( x_i \) by:

\[
x_i = \mathcal{N}_l \left( C \left[ y_{i-N}^{\text{w}}, \ldots, y_{i-1}^{\text{w}}, y_i^{\text{w}}, y_{i+1}^{\text{w}}, \ldots, y_{i+N}^{\text{w}} \right] \right),
\]

where \( \mathcal{N}_l \) denotes the latent HR frame restoration network; \( y_i^{\text{w}} \) denotes the extracted feature of the \( i \)-th LR frame (i.e., \( y_i^{\text{w}} = \mathcal{N}_e(y_i) \)) and \( y_j^{\text{w}} \) denotes the warped features according to the estimated optical flow (i.e., \( y_j^{\text{w}} = F_{j\rightarrow i} \mathcal{N}_e(y_j) \)). For the network architecture of \( \mathcal{N}_l \), we adopt 20 ResBlocks [33] for feature reconstruction and use the pixel shuffle modules [34]...
Fig. 2. Overview of the proposed method. The proposed self-supervised learning-based deep CNN model contains two branches. The main branch is used to estimate blur kernel, the optical flow, and latent HR frame under the self-supervision of the LR input frame. The auxiliary branch uses the auxiliary paired data, which is generated based on the LR input frames and estimated blur kernel from the main branch, to constrain the network training for the optical flow and latent HR frame. The video super-resolution module (VSR) in these two branches share the same network parameters. All the branches are jointly trained in an end-to-end manner based on the self-supervised learning method. Please refer to the main content for details.

for upsampling. The detailed network configurations of $N_I$ are shown in Fig. 3.

D. Self-Supervised Learning

As the ground truth HR videos and blur kernels are not available, a straightforward way to train the networks $N_k$, $N_f$, $N_e$, and $N_I$ is to minimize the following loss function:

$$L_{self} = \rho(SK_i x_i - y_i), \quad (5)$$

where $\rho(\cdot)$ denotes a robust function. We use the $L_1$-norm in this paper. However, directly minimizing (5) has infinite solutions, and the trivial solutions are likely to be obtained in most cases [25], [35]. To overcome this problem, we explore the properties of the blur kernels and the image formation model (1) to constrain the blur kernel estimation and latent HR frame restoration process.

As the elements of the blur kernels are usually sparse, we develop a hyper-Laplacian prior to model the sparse property of the output of the network $N_k$:

$$L_k = \|K_i\|^\alpha, \quad (6)$$

where $\alpha$ denotes a hyperparameter whose value is usually taken 0.5 according to [3].

To regularize the latent HR frame restoration process, we develop a video degradation constraint based on the image formation model (1). Before presenting the video degradation constraint, we first introduce the following property.

**Property:** Let $K_i^e$, $F_{i \rightarrow j}^e$, and $F$ denote the ground truth blur kernel matrix, warping matrix, and the exact LR-to-HR mapping function. That is, the following equation

$$x_i = F(y_{i-N}; \ldots; y_{i-1}; y_i; y_{i+1}; \ldots; y_{i+N}) \quad (7)$$

strictly holds. Therefore, for any videos $\{L_j\}$, if $L_j = SK_jF_{i \rightarrow j}^eH_i$, we have:

$$H_i = F(L_{i-N}; \ldots; L_{i-1}; L_i; L_{i+1}; \ldots; L_{i+N}). \quad (8)$$

In this property, the exact LR-to-HR mapping function is an ideal function that can accurately recover latent HR frames from degraded LR frames, and it is independent of video content, so it can hold for any video with the same degradation parameters. In addition, we assume that the motion field is accurately estimated in advance according to any specifically given high-resolution video $\{H_i\}$. As the blur kernels are independent from the video contents, we can generate auxiliary LR video $\{L_i\}$ by applying the estimated blur kernel to any HR video $\{H_i\}$ according to the image formation model (1). Instead of using additional HR reference videos, we directly use $\{y_i\}$ as the HR video to generate auxiliary LR video $\{L_i\}$. Based on the above property, if the latent HR frame restoration network $N_I$ is accurately estimated, the output of the network $N_I$ should be close to $y_i$.

Thus, we develop a constraint to regularize the network $N_I$ by:

$$L_I = \rho(N_I(C[L_{i-N}^{e,m}, \ldots; L_{i-1}^{e,m}, L_i^{e,m}, L_{i+1}^{e,m}, \ldots; L_{i+N}^{e,m}]) - y_i), \quad (9)$$
Fig. 3. Detailed network configurations of the blur kernel estimation network \(N_k\), the feature extraction network \(N_e\), and the latent HR frame restoration network \(N_I\). The blur kernel estimation network \(N_k\) takes the input sequence \(\{y_{i-2}, y_{i-1}, y_i, y_{i+1}, y_{i+2}\}\) as input and outputs the estimated blur kernel \(K_i\).

The feature extraction network \(N_e\) extracts the feature \(y_{e,w}^i\) from the input frame \(y_i\). The latent HR frame restoration network \(N_I\) takes the aligned features \(\{y_{i-2}^{e,w}, y_{i-1}^{e,w}, y_i^{e,w}, y_{i+1}^{e,w}, y_{i+2}^{e,w}\}\) as input and recovers the latent HR frame \(x_i\).

where \(L_e^i\) denotes the extracted feature of the auxiliary LR frame \(L_i; \{L_{j}^{e,w}\}_{j=1}^{N_j+1} \subseteq \mathbb{R}^{N_j \times 3 \times 3}\) denotes the warped features according to the estimated optical flow \(N_f(L_j, L_i)\) and \(\{L_i\} = \{SK_i y_i\}\).

Based on the above considerations, the proposed self-supervised learning for the video SR can be achieved by minimizing:

\[
L = L_{self} + \lambda L_I + \gamma L_k,
\]

where \(\lambda\) and \(\gamma\) are weight parameters. We will demonstrate the effectiveness of self-supervised learning in Section V-A.

When the training stage is done, only the VSR network (i.e., \(N_f\), \(N_e\), and \(N_I\)) is used at the inference stage to recover the latent HR videos.

E. Implementation Details and Datasets

1) Implementation Details: We use the pre-trained model from [31] as the initialization of the optical flow estimation network \(N_f\), and randomly initialize the networks \(N_k\), \(N_e\), and \(N_I\). After initialization, all parameters are jointly trained under the proposed self-supervised constraints. The learning rate of networks \(N_k\), \(N_e\), and \(N_I\) is initialized to \(10^{-4}\), and the learning rate of network \(N_f\) is initialized to \(10^{-6}\) as it adopts the pre-trained model. The ADAM optimizer [36] with parameters \(\beta_1 = 0.9\), \(\beta_2 = 0.999\), and \(\epsilon = 10^{-8}\) is used for the network training. The half length of the input LR sequence \(N\) is set to be 2. All the networks are jointly trained in an end-to-end manner based on the self-supervised learning method. The learning rate values decrease to half after every 100 epochs, and 200 epochs are used. For the weight parameters in the self-supervised loss function (10), we empirically set \(\lambda\) and \(\gamma\) to be 1 and 0.04. Similar to [3], we further use the boundary loss and the center loss to constrain the estimated blur kernels by encouraging the boundary values to be zero and the center of mass to be at the center of the kernel, and their weights are set to be 0.5 and 1.0 respectively. The training code, models and experimental results used in the paper will be available at https://github.com/csbhr/Self-Blind-VSR.

2) Datasets: We first evaluate the proposed algorithm on the synthetic datasets, which are generated with the different Gaussian blur kernels and the realistic motion blur kernels [3] based on the image formation model (1). Then, we further
To generate the synthetic dataset for quantitative evaluation, the REDS dataset [37] is used as the training dataset, and the commonly used video SR datasets the REDS4 dataset (split from [37] by [5]), the VID4 dataset [6], and the SPMCS dataset [8] are adopted as benchmarks to evaluate our method. When generating the LR videos, we apply the blurring operation with the downsampling operation to each HR video according to the image formation model (1). For the blurring operation, we use the different Gaussian blur kernels and the realistic motion blur kernels from [3]. For the Gaussian blur kernels, the standard variation values range from 0.4 to 2. The downsampling operation is used to extract the pixels based on the scale factor, and the scale factor is set to be 4.

IV. EXPERIMENTAL RESULTS

As our method aims to solve the blind video SR problem in a self-supervised manner, few methods have been proposed for this problem. To evaluate the performance of our algorithm, we still compare it against the related self-supervised single image SR methods (ZSSR [2], KernelGAN [3], MZSR [38]), the supervised single image SR method (RCAN [23], IIC [1], RBPN [4], DUF [14], TDAN [15], EDVR [5]), BasicCSR [39], BasicCSR++ [40], RVRT [41]). For ZSSR [2], the default kernels are Bicubic ones for evaluations. For KernelGAN [3], we use the released codes with the default settings, where ZSSR [2] is used to restore HR images for evaluations.

For the comparisons with the supervised learning-based methods, as the ground truth HR videos are not available for training, the supervised SR methods (RCAN [23], IIC [1], RBPN [4], DUF [14], TDAN [15], EDVR [5], BasicCSR [39], BasicCSR++ [40], RVRT [41]) are evaluated with the officially provided pre-trained models for fairness. We use the PSNR and SSIM as the quantitative evaluation metrics for the synthetic videos.

A. Quantitative Evaluations

Table I–III show the quantitative evaluations on the benchmark datasets with different Gaussian blur kernels, where our method generates the results with the highest PSNR and SSIM values than state-of-the-art methods. We note that the RCAN [23] method does not generate high-quality SR results as it assumes that the blur kernel is known. Although the blind image SR methods [1], [3] involve the blur kernel estimation, they are designed for single images and do not perform well on the video SR problems. It shows that the PSNR values of our method are at least 1.06 dB higher than these blind image SR methods. For the non-blind video SR methods [4], [5], [14], [15], [39], [40], [41] which are based on supervised learning approach, as they assume that the blur kernel is known and train deep models on the datasets with fixed degradation (e.g., Bicubic downsampling), they do not perform well on the datasets with unknown blur kernels under blind video SR settings. In contrast, our method explicitly involves the blur kernel estimation and solves the deep models in a self-supervised manner. Thus, it can generate favorable results against these video SR methods without any paired or unpaired training datasets.

We further evaluate our method when the blur kernels are complex realistic ones [3] in Table IV. We note that the methods [2], [3], [38] based on the self-supervised learning algorithm, the supervised image SR [1], [23] and the supervised video SR methods [4], [5], [14], [15], [39], [40], [41] do not generate good

| Methods | Bicubic | RCAN [23] | MZSR [38] | ZSSR [2] | KernelGAN [3] | IIC [1] | RBPN [4] | DUF [14] | TDAN [15] | EDVR [5] | BasicCSR [39] | BasicCSR++ [40] | RVRT [41] | Our |
|---------|---------|----------|-----------|----------|----------------|--------|---------|---------|---------|---------|---------------|---------------|----------|-----|
| PSNR    | 22.27   | 23.13    | 23.99     | 22.83    | 16.07          | 23.80  | 22.07   | 23.04   | 23.94   | 23.82   | 23.34         | 24.19         | 23.45    | 28.59 |
| SSIM    | 0.4313  | 0.7808   | 0.6489    | 0.6455   | 0.5403         | 0.8955 | 0.6927  | 0.7522  | 0.7857  | 0.7696  | 0.7525        | 0.7956        | 0.7965   | 0.7829 |

The proposed method generates the results with the highest values.

TABLE II

| Methods | Bicubic | RCAN [23] | MZSR [38] | ZSSR [2] | KernelGAN [3] | IIC [1] | RBPN [4] | DUF [14] | TDAN [15] | EDVR [5] | BasicCSR [39] | BasicCSR++ [40] | RVRT [41] | Our |
|---------|---------|----------|-----------|----------|----------------|--------|---------|---------|---------|---------|---------------|---------------|----------|-----|
| PSNR    | 23.51   | 23.77    | 24.43     | 25.15    | 15.07          | 24.74  | 24.14   | 25.00   | 25.50   | 25.27   | 27.73         | 27.53         | 27.11    | 27.77 |
| SSIM    | 0.7241  | 0.8694   | 0.7978    | 0.7968   | 0.3501         | 0.7920 | 0.7942  | 0.8129  | 0.8149  | 0.8055  | 0.7985        | 0.8002        | 0.7957   | 0.8184 |

The proposed method generates the results with the highest values.

| Methods | Bicubic | RCAN [23] | MZSR [38] | ZSSR [2] | KernelGAN [3] | IIC [1] | RBPN [4] | DUF [14] | TDAN [15] | EDVR [5] | BasicCSR [39] | BasicCSR++ [40] | RVRT [41] | Our |
|---------|---------|----------|-----------|----------|----------------|--------|---------|---------|---------|---------|---------------|---------------|----------|-----|
| PSNR    | 23.51   | 23.77    | 24.43     | 25.15    | 15.07          | 24.74  | 24.14   | 25.00   | 25.50   | 25.27   | 27.73         | 27.53         | 27.11    | 27.77 |
| SSIM    | 0.7241  | 0.8694   | 0.7978    | 0.7968   | 0.3501         | 0.7920 | 0.7942  | 0.8129  | 0.8149  | 0.8055  | 0.7985        | 0.8002        | 0.7957   | 0.8184 |

The proposed method generates the results with the highest values.
TABLE IV

Comparisons of the Video SR Results by the State-of-the-Art Methods on the REDS4 dataset [5] With Realistic Motion Blur Kernels [3] in Terms of PSNR and SSIM

| Methods     | Bicubic | RCAN [20] | MZSR [38] | ZSSR [2] | KernelGAN [3] | IFC [1] | RRPN [4] | DLU [14] | TDAN [15] | EDVR [5] | BasicVSR [39] | BasicVSR++ [40] | RVET [41] | Ours |
|-------------|---------|-----------|-----------|----------|--------------|---------|---------|---------|---------|---------|--------------|--------------|---------|------|
| PSNR        | 25.76   | 24.84     | 24.67     | 24.38    | 24.66       | 24.83   | 24.51   | 27.13   | 27.27   | 27.64   | 27.74        | 27.82        | 29.44   |      |
| SSIM        | 0.7937  | 0.7939    | 0.7939    | 0.7903   | 0.7957     | 0.7978  | 0.7972  | 0.7979  | 0.8084  | 0.7994  | 0.8057       | 0.8104       | 0.8372  |      |

The proposed method generates the results with the highest values.

Fig. 4. Comparison of the video SR results on the REDS4 dataset [5] (×4). Our method recovers high-quality frame with clearer structures.

results. In contrast, our method generates the results with higher PSNR and SSIM values, which demonstrates that our method generalizes well compared to the existing methods.

B. Qualitative Evaluations

Fig. 4 shows some visual comparisons of SR results generated by the evaluated methods on the REDS4 dataset [37] with a scale factor of 4. We note that the ZSSR method [2] does not generate a clearer image as it does not involve the blur kernel estimation (Fig. 4(e)). Although the KernelGAN method [3] explicitly estimates blur kernels from LR images and can use the ZSSR method to super-resolve images, the generated results contain significant artifacts due to the imperfect blur kernels (Fig. 4(f)). To correct errors of the blur kernels, the IKC method [1] develops an effective iterative kernel correction method. Although the quality of the SR results is improved (Fig. 4(d)), the structural details are still not restored well as this method is not designed for the video SR problem. Although the RBPN method [4] and EDVR method [5] are developed to solve video SR, these methods assume that the blur kernel is known and do not solve the blind video SR problem well as shown in Fig. 4(g) and (h). In contrast, although our method does not require the HR videos as the supervision, it generates much clearer frames with better structural details as shown in Fig. 4(i). This further demonstrates the effectiveness of the proposed method on the blind video SR problem.

Figs. 5 and 6 show some examples from the VID4 dataset [6] and the SPMCS dataset [8]. Our method generates the frames with finer details, where the characters in the restored frames are recognizable. In addition, Fig. 7 shows the visual comparisons on the REDS4 dataset [5] with the realistic motion blur kernels from [3]. We note that the ambiguous directions of building boundaries due to the downsampling operation fails these state-of-the-art methods, while our method generates better results with correct directions of building boundaries.

We further evaluate our method on real-world videos, and show the visual results in Figs. 8–10. We note that
state-of-the-art methods do not restore the structural details well, where the restored results are still blurry or contain significant artifacts. In contrast, our algorithm generates much clearer frames. More comparisons on real-world videos can be found in supplemental material.

V. ANALYSIS AND DISCUSSIONS

In this section, we provide further analysis of the proposed self-supervised learning method and discuss the major differences from the closely-related methods.
To demonstrate the effectiveness of the proposed self-supervised learning, we disable the constraint (9) and (6) in the proposed method for fair comparisons. Table V shows the quantitative evaluations on the benchmark datasets. We note that the method without the constraint (9) and (6) does not generate high-quality videos. The PSNR values of this baseline method are at least 2.54 dB lower than those of the proposed method, suggesting the effectiveness of the proposed self-supervised learning on the video SR problem. Fig. 11(d) shows the estimated SR results by the method without the constraint (9) and (6), where there exist significant artifacts in the restored images. In addition, the estimated blur kernel by the method without the constraint (9) and (6) looks like a delta kernel (see Fig. 13(a)), which further verifies our claims in Section III-D.

Since the blur kernels involved in LR image acquisition are more like low-pass filters, it is necessary to verify whether the proposed sparsity constraint (6) works effectively. In [3], it has
Table VI

| Methods       | w/o temporal | w/ temporal (Ours) |
|---------------|--------------|-------------------|
| PSNR          | 28.40        | 29.23             |
| SSIM          | 0.8246       | 0.8433            |

been proven to be effective on low-pass filters, and here we further quantitatively evaluate the constraint (6). Table V and Fig. 11(e)&(f) show that without using (6) does not perform well. Moreover, Table VII shows the quality of the regenerated LR videos by using the estimated blur kernels, where the PSNR value of our method is 5.67 dB higher than the baseline method that disables the constraint (6) (i.e., “w/o (6)” in Table VII). This indicates that better blur kernels are estimated for the construction of the auxiliary supervised constraint. The above results demonstrate that although constraint (6) is in favor of sparse blur kernels, it can work well on low-pass filters when used together with data fidelity terms (i.e., constraint (5) and (9)).

In addition, we note that the method only using the constraint (5) and (6) (i.e., “w/o (9)” in Table V) does not generate good results, where the PSNR values of this baseline are at least 2.10 dB lower than that of the proposed method. This demonstrates that using the proposed auxiliary supervised constraint (9) is able to help video SR when the ground truth videos are not available.

We further verify whether arbitrary blur kernels can also make the proposed self-supervised training effective or not. We replace the estimated blur kernels with two kinds of blur kernels: 1) delta kernels that make the blurring process absent; 2) ground-truth kernels that act as an upper-bound. We retrain these two baselines using the same settings as our method. We note that the method using delta kernels (i.e., “w/ delta kernels” in Table V) does not perform well, where the PSNR values of this baseline are at least 2.59 dB lower than those of our method. In contrast, our method that uses the estimated blur kernels is able to generate comparable results compared to the upper-bound method using ground-truth kernels (i.e., “w/ GT kernels” in Table V). This suggests that the estimated blur kernels are crucial for the network training, and using arbitrary blur kernels cannot make self-supervised learning effective.

We note that the downsampling operation in the constraint (5) discards most of the pixels in x_i, which means that only a small part of the pixels in x_i can be constrained. This may prevent the network N_f from learning correctly when the ground-truth constraint is not available. In addition, (5) constrains the blur kernel estimation network N_k and the latent HR frame restoration network N_f at the same time, which may make the two networks compromise with each other and make the training process unstable. As the constraint (9) can effectively regularize the network N_f, we block the constraint (5) from updating the network N_f with a detach operation which is the stop-gradient operation implemented by ‘torch.Tensor.detach’, and the constraint (5) is only used to regularize the network N_k. To demonstrate the effectiveness of this detach operation in (5), we disable this operation and retrain this baseline method (i.e., “w/o detach in (5)” in Table V). It shows that the PSNR values of this baseline method are at least 1.56 dB lower than that of the proposed method, which verifies the necessity of this detach operation in the network training process.

B. Relations With the Unpaired and Self-Supervised Learning Image SR Methods

We note that several methods [2], [3] develop effective self-supervised learning methods based on deep neural networks to solve the single image SR problem. In [2], Shocher et al. develop an effective image-specific CNN model to solve the single image SR problem which is achieved by the zero-shot learning algorithm. In the training process, this method first learns the LR-to-HR mapping function from the coarser-resolution one of the input LR image and then applies the learned mapping function to the input LR image for the latent HR image restoration. This training strategy is similar to our constraint (9). However, the proposed constraint (9) implicitly contains the temporal information which can explore the useful features from adjacent frames for better video SR. Without using the temporal information, the proposed method would not generate better results. To verify this, we use the features of original generated auxiliary LR frames \{L_i^a\} instead of \{L_i^{a-w}\} in (9) for comparisons (“w/o temporal” for short). Table VI shows that directly using the features of original generated auxiliary LR frames \{L_i^a\} instead of \{L_i^{a-w}\} in (9) does not generate good videos. And the comparisons shown in Fig. 12(e) and (f) demonstrate that without using temporal information does not generate clear SR images. This suggests that effectively exploiting the temporal information of multiple input frames is important for blind video SR.

In addition, the ZSSR method does not involve the blur kernel estimation. Although its performance can be significantly improved by using the blur kernels from [3], the super-resolved images would be affected by inaccurate kernels. In contrast, our method explicitly involves the blur kernel estimation, where blur kernel estimation, the optical flow estimation, and video frame restoration are simultaneously solved in a unified framework. Thus, it can better reduce the influence of the inaccurate kernel.
We further note that Maeda [16] develops an effective image SR algorithm based on unpaired data, where the network training does not require the corresponding HR images. This method needs several networks to generate pseudo supervision from additional exemplar datasets to train the networks. However, using more networks will accordingly increase the difficulty of the network training. In contrast, our method does not require any other additional exemplar datasets, and our video degradation constraint does not introduce additional networks, which makes the training process easier than [16]. Moreover, the method by [16] is designed for single image SR, which cannot be extended to the blind video SR problem directly.

C. Accuracy of the Auxiliary Synthesized LR Videos

As we do not have any HR videos as the supervision, we propose to estimate blur kernels and optical flow to explore the effective constraint from LR videos based on the image formation model (1). Thus, the estimated blur kernels are mainly used to generate the synthesized LR videos for constructing the auxiliary supervised constraint (9), which can regularize the deep model for better HR video restoration.

Table VII shows that the quality of the synthesized LR videos by using the estimated blur kernels is better than that of the ones generated by the blur kernels [3]. The results also indicate that the proposed method generates better blur kernels, which is able to effectively model the blurring process in image formation model (the average PSNR value of re-synthesized LR videos is 39.62). Although the proposed method does not focus on the accurate recovery of blur kernels, we still visualize the estimated blur kernels in Fig. 13. It shows that the proposed constraints (9) and (6) is able to improve blur kernel estimation.

D. Online Fine-Tuning on Real Scenarios

As our method does not require ground truth HR videos as the supervision, real-world degraded videos can be used to fine-tune our method for better generalization. Thus, our method can be fine-tuned online. To verify this property, we randomly choose 10 real degraded videos from websites and use them to fine-tune the proposed model in a self-supervised fashion. As the corresponding ground truth HR videos are not available, we use the non-reference metric NIQE [43] to evaluate the proposed method. Table VIII shows that our method generates results with better visual perceptual quality than the supervised video SR methods [5], [40]. Moreover, fine-tuning the pretrained video SR models (including our model and existing video SR models [5], [40]) on degraded videos using the proposed self-supervised approach is able to generate better results on these real scenarios (see “w/ OFT & Est. Ker.” in Table VIII). Fig. 14 shows some visual results, where our method fine-tuning using real degraded videos generates better results with finer structures (see Fig. 14(d)). In addition, to verify whether the performance improvement caused by online fine-tuning is due to the network adapted to the “contents” of the unseen video or not, we use pre-defined blur kernels (i.e., Bicubic kernel) instead of the proposed kernel estimation network to fine-tune the proposed model. Table VIII shows that using predefined blur kernels does not perform well (see “w/ OFT & Bic. Ker.” in Table VIII), suggesting that the performance gain is due to the proposed self-supervised approach, rather than adaptation to the video contents.

E. Evaluations of the Proposed Method Using HR Videos as Supervision

As the proposed method is designed for the blind video SR problem when the ground truth HR videos are not available, it is interesting to examine whether the proposed method works well if the HR videos are used to supervise the training of the proposed network. To this end, we use the LR videos and their corresponding HR videos to train the proposed network, where the commonly used L1-norm based loss function is applied to ensure the network output is close to the HR frame. Table IX shows that the proposed method without using the HR videos as the supervision generates comparable results compared to the method using the HR videos as the supervision on the SPMCS dataset, which further indicates that the proposed self-supervised
learning method is able to solve the blind video SR problem when the ground truth videos are not available.

F. Robustness to Image Noise

We further evaluate our method on the datasets with image noise. We add the Gaussian noise to each LR frame generated in the way mentioned in Section III-E2, where the noise level ranges from 0 to 10%. We train the proposed method on this noisy dataset, and evaluate it on REDS4 datasets where we add the Gaussian noise with the noise level ranging from 1% to 5%. Table X shows that our method generates results with the highest PSNR and SSIM values when the test datasets contain image noise, indicating that the proposed method is robust to noise to some extent.

G. Evaluations of the Proposed Method on the Blind Image SR Problem

To examine whether the proposed self-supervised learning works on the blind image SR problem or not, we remove the optical flow estimation module from the VSR network, and estimate the latent HR frame and the blur kernel from the single input LR image. We retrain this baseline method using the same training settings as the proposed method. Table XI shows that our method generates results with higher PSNR and SSIM values compared to existing image SR methods.

H. Model Size, Running Time, and Computational Complexity Comparisons

We further compare the model complexity of our method with state-of-the-art methods in terms of model parameters, running time, and floating point operations (FLOPs). These metrics are tested on the same machine with an Intel Core i7-7700 CPU and an NVIDIA GTX 1080Ti GPU, where the spatial resolution of input videos is 720 × 1280 pixels. Since the methods MZSR [38], ZSSR [2], and KernelGAN [3] need to update parameters through multiple back-propagations during inference, we do not report their FLOPs values. Table XII shows that our method is more efficient than most of the compared methods.

I. Limitation Analysis and Future Work

In order to avoid the trivial solutions caused by directly minimizing (5) when there are no ground-truth HR videos as supervision, we first explore the sparse property of the estimated blur kernels and then use the estimated blur kernels and original LR input videos to generate auxiliary paired data for constraining the latent HR video restoration process. The analysis in Section V-C has demonstrated that the proposed self-supervised learning can improve the accuracy of the generated auxiliary paired data. However, as shown in Fig. 13(b), there is still a certain gap between the estimated blur kernel and the ground-truth one. Although the proposed method only focuses on the accuracy of the generated auxiliary paired data but not the estimated blur kernels, future work will study how to estimate more accurate blur kernels so that the performance of the proposed self-supervised method can be further improved.

In addition, although the method based on the image formation model (1) can handle some real-world examples to some extent as shown in Figs. 8–10, it does not involve nonlinear degradations, such as camera response functions which usually exist in real-world degradation processes. Future work will study how to extend the proposed self-supervised learning to more complex degradation models considering nonlinear degradation operations.

VI. CONCLUSION

We have proposed an effective video SR method based on a self-supervised learning method and developed a simple and effective method to generate auxiliary paired data from the original LR input videos to constrain the network training. We have introduced an optical flow estimation module to exploit the information from adjacent frames for better HR video restoration. We have shown that our method also can adopt existing self-supervised learning methods and developed a simple and effective method to generate auxiliary paired data from the original LR input videos to constrain the network training. We have shown that our method can handle some real-world examples to some extent as shown in Figs. 8–10, it does not involve nonlinear degradations, such as camera response functions which usually exist in real-world degradation processes. Future work will study how to extend the proposed self-supervised learning to more complex degradation models considering nonlinear degradation operations.
REFERENCES

[1] J. Gu, H. Lu, W. Zuo, and C. Dong, “Blind super-resolution with iterative kernel
estimation using an internal-GAN,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog.-
nit. 2018, pp. 3118–3126.

[12] J. Caballero et al., “Real-time video super-resolution with spatio-temporal
networks and motion compensation,” in Proc. IEEE Conf. Comput. Vis. Pattern
Recog., 2017, pp. 2848–2857.

[21] A. Shocher, N. Cohen, and M. Irani, “Zero-shot super-resolution using
deep internal learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog.-nit. 2018,
pp. 3897–3906.

[14] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, “Deep video super-resolution
network using dynamic upsampling filters without explicit motion compen-
sation,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog., 2018, pp. 3224–3232.

[9] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, “TDAN: Temporally-deformable
alignment network for video super-resolution,” in Proc. IEEE Conf. Comput. Vis.
Pattern Recog., 2020, pp. 291–300.

[10] M. Irani and S. Peleg, “Improving resolution by image registration,”
Graph. Model Image Process., vol. 53, no. 3, pp. 231–239, 1991.

[18] M. Irani, Z. Luo, Y. Huang, S. Li, L. Wang, and T. Tan, “Unfolding the alternating
optimization for blind super resolution,” in Proc. Adv. Neural Inf. Process. Syst.,
2020, pp. 5652–5643.

[15] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, “TDAN: Temporally-deformable
alignment network for video super-resolution,” in Proc. IEEE Conf. Comput. Vis.
Pattern Recog., 2018, pp. 3513–3522.

[35] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding and
evaluating blind deconvolution algorithms,” in Proc. IEEE Conf. Comput. Vis.
Pattern Recog., 2009, pp. 1964–1971.

[5] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, “EDVR:
Video restoration with enhanced deformable convolutional networks,” in
Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops, 2019, pp. 1954–1963.

[20] R. Liao, X. Tao, L. Xu, J. Jia, and E. Wu, “Handling motion blur in
multi-frame super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern
Recog., 2015, pp. 5224–5232.

[4] W. Shi et al., “Real-time single image and video super-resolution using
an efficient sub-pixel convolutional neural network,” in Proc. IEEE Conf. Comput.
Vis. Pattern Recog., 2016, pp. 1874–1883.

[17] M. Irani and S. Peleg, “Improving resolution by image registration,”
IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 2, pp. 109–122, Jun. 1998.

[11] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, “Video super-resolution via
draft-ensemble learning,” in Proc. IEEE Conf. Comput. Vis. Pattern
Recog., 2015, pp. 531–539.

[13] D. Liu et al., “Robust video super-resolution with learned temporal
dynamics,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2526–2534.

[16] J. Caballero et al., “Real-time video super-resolution with spatio-temporal
networks and motion compensation,” in Proc. IEEE Conf. Comput. Vis. Pattern
Recog., 2017, pp. 2848–2857.

[8] J. Pan, H. Bai, J. Dong, J. Zhang, and J. Tang, “Deep blind video super-
resolution,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 4811–4820.

[19] Q. Shan, Z. Li, J. Jia, and C. Tang, “Fast image/video upsampling,”
in Proc. CVPR, 2017, pp. 3118–3126.

[28] A. Bulat, J. Yang, and G. Tzimiropoulos, “To learn image super-resolution,
use a GAN to learn how to do image degradation first,” in Proc. Eur. Conf.
Comput. Vis., 2018, pp. 187–202.

[22] J. Gu, H. Lu, W. Zuo, and C. Dong, “Blind super-resolution with iterative
kernel correction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2019,
pp. 1604–1613.

Haoran Bai is working toward the PhD degree with the School of Computer Science and Engi-
nering, Nanjing University of Science and Tech-
nology, China. His research interests include im-
agedeblurring, dehazing, and other restoration tasks.

Jishan Pan (Senior Member) received the PhD de-
gree in computational mathematics from the Dalian
University of Technology, China, in 2017. He was a joint-training PhD student with the School of Math-
ematical Sciences, Dalian University of Technology,
China, and Electrical Engineering and Computer Sci-
ence, University of California, Merced, CA. He is a
professor with the School of Computer Science and
Engineering, Nanjing University of Science and Tech-
nology. His research interest includes image de-
blurring, image/video analysis and enhancement, and
related vision problems.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.