IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 25, 2021, accepted July 4, 2021, date of publication July 19, 2021, date of current version August 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098326

Efficient Video Super-Resolution via Hierarchical Temporal Residual Networks

ZHI-SONG LIU[1,2], (Member, IEEE), WAN-CHI SIU[1,2], (Life Fellow, IEEE), AND YUI-LAM CHAN[1], (Member, IEEE)

[1]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong
[2]Caritas Institute of Higher Education, Hong Kong

Corresponding author: Wan-Chi Siu (enwcsiu@polyu.edu.hk)

This work was supported in part by the Hong Kong Polytechnic University Internal Research Grant ZZHR, and in part by the RGC Project of the Hong Kong Special Administrative Region, China, under Grant University Grants Committee (UGC)/IDS(C)11/E01/20.

ABSTRACT Super-Resolving (SR) video is more challenging compared with image super-resolution because of the demanding computation time. To enlarge a low-resolution video, the temporal relationship among frames must be fully exploited. We can model video SR as a multi-frame SR problem and use deep learning methods to estimate the spatial and temporal information. This paper proposes a lighter residual network, based on a multi-stage back projection for multi-frame SR. We improve the back projection based residual block by adding weights for adaptive feature tuning, and add global & local connections to explore deeper feature representation. We jointly learn spatial-temporal feature maps by using the proposed Spatial Convolution Packing scheme as an attention mechanism to extract more information from both spatial and temporal domains. Different from others, our proposed network can input multiple low-resolution frames to obtain multiple super-resolved frames simultaneously. We can then further improve the video SR quality by self-ensemble enhancement to meet videos with different motions and distortions. Results of much experimental work show that our proposed approaches give large improvement over other state-of-the-art video SR methods. Compared to recent CNN based video SR works, our approaches can save, up to 60% computation time and achieve 0.6 dB PSNR improvement.

INDEX TERMS Video, deep learning, residual network, hierarchical structure, super-resolution.

I. INTRODUCTION

The advent of high-definition and ultra-high-definition television demands rapid development of image and video processing in various applications. One of the applications is video super-resolution (SR). Given the fact that the 4K or 8K resolution video contains millions of pixels, it is not only difficult for broadcast but also for storage. To suit for high-definition devices, video SR is useful for enlarging low resolution (LR) video to high resolution (HR) video with good visual quality.

Recently, various methods have been proposed to resolve the video SR problem. Based on whether using the temporal information or not, we can classify the video SR methods into two categories: single image SR [1]–[26] and multi-frame SR [27]–[44], [51]–[58].

For single image SR, the task is to consider a video containing independent frames and super-resolve the video frame by frame. Based on the methods to resolve the SR problem, researchers can either use traditional machine learning approaches [1]–[8] to tackle the image SR as patch based restoration or deep learning based restoration [9]–[26] to learn nonlinear regression models. For traditional machine learning based approaches, the key is to approximately model the nonlinear image SR as an ensemble of linear patch based SR. Hence, we can use classification method, such as kNN [1], [2], Random Forest [7], [8] to classify patches into groups and then each group can learn one linear regression model to map the LR patches to the corresponding HR patches. However, this divide-and-conquer strategy introduces information loss due to possible wrong classifications. To reduce the variance of mean squared errors between LR and HR patches, much computation time must be used to search for fine classification for reconstruction for most of these approaches.

Instead of learning more and more complex features for classification, deep learning based methods simply design an

The associate editor coordinating the review of this manuscript and approving it for publication was K. C. Santosh.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

end-to-end Convolutional Neural Network (CNN) structure to directly learn the nonlinear relationship between LR and HR images. The advantage is that deep learning approaches usually train huge datasets to extract global feature maps from the entire image rather than patches. Straightforward CNN based models for single image SR is one of the popular approaches [9]–[26]. Inspired by the success of ResNet [28] in image visual recognition, [11], [14], [15] propose more modifications on the residual learning in image SR. The approach resonates with the assumption that LR and HR images share the same low frequency information. Back projection (BP) has been recently introduced into the deep convolutional networks, the result of which has been proven to be very efficient in image SR [16], [17]. Most recently, attention is widely used to learn nonlocal features for image SR [20]–[26], including channel attention [20], [22] spatial attention [21] and so on [23]–[26]. Both attention models were designed for image SR that are efficient enough for video SR in real-time applications.

Despite the good performance of single image based SR, it does not fully exploit the temporal correlation for video super-resolution. The reconstruction of the spatial domain could be compensated by temporal correlation among neighborhood frames. However, to process multiple frames, there are some key issues that need to be resolved: 1) the high redundant information in neighborhood frames makes the training process very difficult to process; 2) the deeper and more complicated CNN structures can achieve better visual quality but cannot properly handle the temporal relationship for real-time implementation, and; 3) the concept of CNN is to extract the spatial correlation which is a big limitation for its uses in video SR. To resolve these problems, [30]–[34], [51]–[55] propose various approaches to extract the temporal information by joint neighborhood frames training. Performance of the video SR has also achieved great improvement. However, most existing video SR [51]–[55] approaches still require pre-motion estimation, like optical flow, to estimate the temporal correlation first and then embed this temporal information with the frames for end-to-end training. Besides the extra computation on motion estimation, the errors caused by motion estimation can be accumulated for the final SR reconstruction.

To achieve a better visual quality as well as using less computation cost, we propose a Space-Time Convolutional Neural Network for video SR (ST-CNN), which is to convolute adjacent frames with the learned filters across both space and time domains to extract the correlation information. The proposed ST-CNN network uses the back projection based residual blocks as a backbone for deep feature extraction. Without any pre-process or extra motion estimation, ST-CNN can learn both the spatial and temporal information using the proposed Spatial Convolution Packing (SCP) scheme. The basic idea is to replace the conventional 2D convolution with a combination of full and partial 2D convolution processes. Interestingly, in this case, we can use fewer parameters to model the long-term intrinsic correlation information across time and spatial domains for video SR.

Meanwhile, we propose to use a many-to-many scheme to train the ST-CNN network. Compared with other video SR methods, instead of using several neighborhood frames to output one SR frame, our proposed ST-CNN can output the same number of SR frames as the input. This multi-input-multi-output can save a significant amount of computation time to achieve fast realization. We also suggest adaptively boosting up the video quality by frame overlapping (ST-CNN(F+)) and patch overlapping processes (ST-CNN(P+)) based on different motions of video sequences to compensate for the error caused by dynamic motions and complex features. We have also designed experiments not only evaluating the performance by comparing the proposed SR with the state-of-the-art video SR methods, but also giving comprehensive and visualized interpretation to the trained ST-CNN model.

Our main contributions can be summarized as follows.

- We propose a Space-Time Convolutional Neural Network, in which we have improved the residual block by adding an adaptive weighting convolution process and global & local connections to learn deep feature representation for video SR.
- The convolution in ST-CNN is done by our proposed Spatial Convolution Packing (SCP) scheme. We can then use fewer parameters and less computation to combine the channel information with the spatial domain for joint training.
- Based on the proposed hierarchical residual network, our ST-CNN can efficiently super-resolve video with fast computation and high visual quality without any pre- or post-process with feature sharing and weighting estimation.
- To boost up the visual quality, we also combine the frame overlapping and patch overlapping processes to form ST-CNN as the proposed ST-CNN+ to further enhance the SR quality in terms of PSNR.

II. RELATED WORK

In this section, let us review the related work from the following perspectives.

Image SR: In the previous research, image SR is regarded as one of the image restoration problems. Given a degraded LR image with various blurring effects and noises, to predict an HR image with rich edges and textures is an ill-posed problem. Developed from local polynomial data mapping, manifold learning techniques are widely studied in the past decades. The key to resolving the image SR is to treat images as an ensemble of redundant patches. The repetitive patch pattern can be learned by classification methods, e.g., Random Forest [7], [8]. Image SR can then be resolved as a patch based classification. However, most, if not all, patch based approaches suffer from the dilemma of model complexity,

that is, the larger patch for training, the harder it for linear approximation.

CNN based methods have been investigated in many computing vision fields and proven to be very competitive compared with traditional machine learning based methods, for their abilities to compute huge datasets and learn nonlinear mappings. There are many research works on image SR [16]–[26], for which they make use of the research findings from early work to design the convolution structure to achieve better SR performance. Residual learning, [11], [14], [15] back projection [16], [17] and attention [20]–[26] are the three most effective processes that are widely used in image SR.

Video SR: In general, video SR can be regarded as multi-frame SR which uses more than one single image to explore the motion information among frames for SR. Related to multi-frame SR, [31] gives an early study on using affine transform to model the motion differences between frames. Benefited from the study of motion estimation in video coding, action recognition and so on, optical flow [29] is widely used in video SR. [31]–[34] extend the optical flow to video SR, making use of motion compensation to generate a SR image with good quality. However, the computation cost of motion estimation is too high for real-time applications. Much research work has been reported to avoid motion estimation in SR. [31] uses Contourlet transform to learn an overcomplete dictionary in the transform domain and search for the optimal reconstruction. Instead of using discriminative learning models, there are also some generative learning approaches proposed to model the motions of the adjacent frames to perform SR. [32] uses Generalized Gaussian Markov Random Fields to model the HR image to preserve the edges and textures. To further improve the frame quality, [33], [34] use Bayesian Maximum a Posteriori (MAP) to estimate the deblurring kernels and motion parameters. Due to the simplified Bayesian model, it achieves fast realization but relatively poor visual quality.

Similar to image SR, video SR can also be resolved by CNN deep learning. [35] firstly makes use of CNN to learn the nonlinear mapping model among the ensembled SR drafts to generate the final SR result. [36] directly uses consecutive frames as input to a CNN network to output SR frames. Considering the difficulty of creating a large video dataset, it uses large image datasets to pre-train the CNN model and then uses a small video dataset to fine-tuning for video SR. Lately, a sub-pixel CNN layer is proposed in [37] to aggregate the feature maps from LR space to pixel-wise reconstruction for efficient image and video SR. [38] further studies the sub-pixel process and proposes an end-to-end CNN model to fuse multiple frames and estimates motion vectors to reveal clear image details. [39], on the other hand, sticks to use a more complex CNN model to learn the temporal dynamics and adaptively aggregate adjacent frames based on the temporal dependency. Recently, there are several works [51]–[55] proposed to use optical flow as the extra information. By making use of mature optical flow packages, both neighborhood frames and optical flow are fed into the network for joint spatial and temporal feature extraction. For example, [51]–[53] propose to directly input neighborhood frames and optical flows for super-resolution via 2D convolution. [54], [55] propose deformable convolution operations for flexible subpixel motion estimation. However, these CNN works are restricted by the concept of the 2D convolution in the spatial domain. A few pieces of works study the joint temporal and spatial feature extraction specifically for video SR [37], [40]–[44], [57], [58], i.e., group convolution and recurrent convolution. One of the straightforward ways is 3D convolution. The 3D convolution was initiated in 2013 [40] for human action recognition. There are also some research findings on using a 3D CNN network for computer vision. Inspired by the success of 3D CNN or pseudo 3D CNN in [40]–[43], there are also some works using 3D CNN for video SR. [37] and [44] make use of the 3D convolution concept to extract the temporal information along with spatial information to perform video SR. Their good performance on video SR proves that the 3D convolution has good potential to be applied to video SR.

In this paper, we come up with an efficient 3D convolution network (ST-CNN) to explore the spatial-temporal dependency for video SR. In the previous study [26], we only discussed the hierarchical residual network in image SR to obtain superior visual quality and computation cost reduction. In this paper, we develop a new study of Spatial Convolution Packing (SCP) and come up with better analysis and efficient model estimation. 1) How does the SCP work in video SR? 2) What is the dependency among the neighborhood frames and how does the short- and long-term temporal information be traded off in video SR? Besides, we have performed many experiments to verify the effectiveness of our model as follows. 1) We will compare our proposed methods with other state-of-the-art video SR approaches on standard video sets in terms of PSNR and computation times with different resolutions. 2) We will use 4K videos to demonstrate the real-world applications of video SR. 3) We will visualize the trained ST-CNN parameters to analyze the physical meanings of spatial and temporal convolution and residual update at different stages of the back projection.

III. HIERARCHICAL TEMPORAL RESIDUAL NETWORK FOR VIDEO SUPER-RESOLUTION

A. FORMULATION

Consider a low-resolution video sequence with additive noise. The objective of video super-resolution is to obtain a high-resolution clear video. Without any pre- and post-processing, our proposed Space-Time Convolutional Neural Network (ST-CNN), as shown in Fig. 1, takes $(2n + 1)$ LR frames as input and generates $(2n + 1)$ SR frames with the desired resolution simultaneously, where n is a positive integer. The network composes of cascaded up- and down-sampling units in the form of up-down-up sampling units to estimate the low- and high- resolution feature maps for SR.

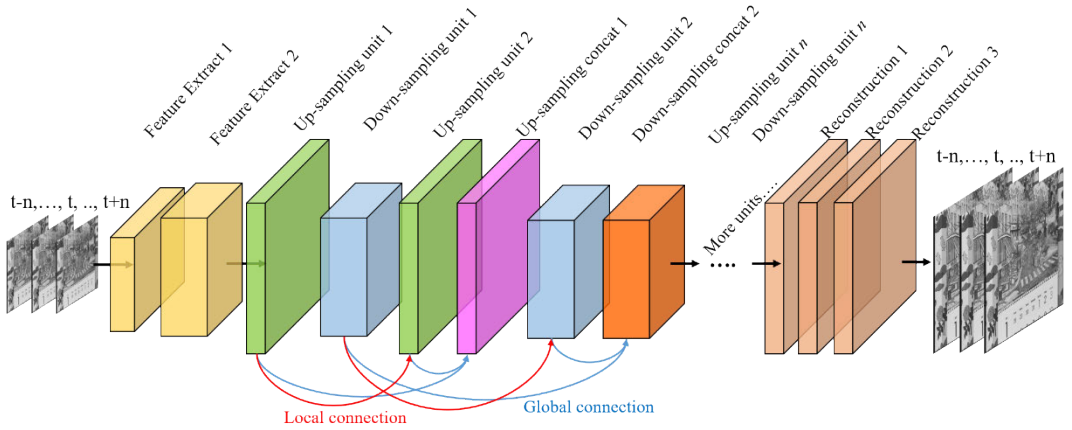

FIGURE 1. The proposed Space-Time Convolutional Neural Network for video SR (ST-CNN) for video SR. It takes $(2n+1)$ LR frames and simultaneously output $(2n+1)$ SR frames. It consists of 2 layers of feature extraction, n layers of up-sampling and down-sampling units, and 3 layers of reconstruction.

We can stack more up-down-up sampling units to explore deeper feature representation. Within each up- and down-sampling unit, instead of using the classical 2D full convolution, we use the proposed SCP scheme (as shown in Fig. 2) to extract 3D features across the spatial and temporal domains for SR. Overall, let us denote LR frames as $\boldsymbol{X}^{LR} \in \mathbb{R}^{H\times W\times C}$ and HR frames as $Y^{HR} \in \mathbb{R}^{\alpha H\times \alpha W\times C}$, where H, W and C are the height, width and number of LR frames, and $\alpha > 1$ is the up-sampling factor. Our proposed ST-CNN network contains three basic stages:

1) Feature extraction. This operation contains two Spatial Convolution Packing (SCP) based convolution layers (''Feature extract 1 & 2'' in Fig. 1) to expand the input LR frames into a larger feature space to increase the dimension of freedom for complex nonlinear mapping.

2) Hierarchical residual update. This operation stacks multiple modules of up-sampling/down-sampling units (''up- & down-sampling units in Fig. 1'', more details about their inside structures are given in Section 2.2.2) to update the residue between LR and HR feature maps. It is the key stage to minimize the loss between SR and ground truth HR frames. Within each unit, we use the proposed SCP scheme for 3D convolution to extract both spatial and temporal information for reconstruction.

3) Final reconstruction. This operation (''Reconstruction 1, 2 & 3'' in Fig. 1) aggregates the intermediate results from the second stage and uses two more 2D full convolution layers to learn the weighted mapping to generate final SR frames.

Fig. 1 shows an overview of the proposed ST-CNN network for video SR, and Fig. 2 shows the proposed SCP scheme for 3D convolution. Let us discuss the details in the following sections.

B. SPATIAL CONVOLUTION PACKING (SCP) PART I

In video SR, temporal information plays a crucial role, which includes the representation of the temporal correlation

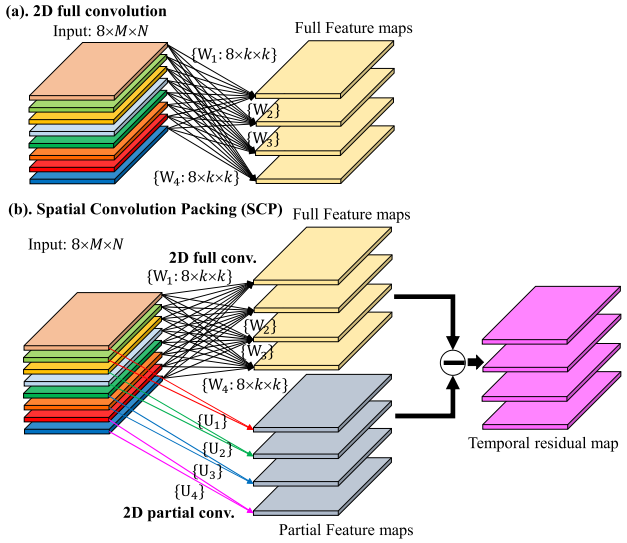

FIGURE 2. Proposed Spatial Convolution Packing (SCP) scheme. where Wi and Ui is the full and partial convolution kernel and k is the kernel size.

dependency between frames. There are two directions to solve this problem. 1) Prior motion estimation: The temporal information is used as an additional feature besides the LR frames. Traditional learning approaches use geometric transform to estimate the motion among frames and use it as data compensation for SR. Similarly, in CNN based video SR, some researchers also input LR frames and the prior motion information to predict the HR frames. However, compared with hand-crafted features, CNN has been proven to be effective to learn features automatically. Using hand-crafted motion estimation for video SR could just form suboptimal estimation. 2) Learnable motion estimation: Naturally, 3D convolution can be applied to video-based processing for its ability to introduce the temporal correlation. In 3D CNN based methods [40]–[43], each input part convolves with a 3D

filter across the spatial and channel domain. Due to one extra domain of freedom, 3D CNN needs a lot of parameters to cover the whole data space and the optimization becomes too complex to find the convergence. Experimental results in [41] indicate that the number of parameters of 3D CNN must be increased dramatically to achieve comparable performance as 2D CNN.

No matter how deep the network, once we input multiple frames to the model, we assume that the temporal correlation among different frames is never lost. It is just hidden in the channel domain of the convolutional feature maps. Instead of explicitly discovering the temporal information among frames, we propose a SCP convolution scheme to perform global and local 2D convolutions. Hence their difference becomes the residual channel correlation among feature maps, and all these can be directly included for training. Fig. 2(a) shows the structure of a conventional 2D full convolution and Fig. 2(b) shows our proposed SCP. As shown in Fig. 2(b), the SCP includes the conventional 2D full convolution (Fig. 2(a)) and partial convolution processes, where we use $\boldsymbol{W}_i$ and $\boldsymbol{U}_i$ ($i = 1, 2, 3, 4$) to represent the 2D full convolution and partial convolution kernels, k is the kernel size and (M, N) is the size of the input. Numerically, in Fig. 2(b), we have input data with the dimension of $8 \times M \times N$ where 8 is the number of channels and (M, N) is the size of input data. For the full convolution, we have 4 kernels $\{\boldsymbol{W}_i\}$ of size $8 \times k \times k$ to convolute with the input data and output 4 feature maps. For the partial convolution, if we group the input data into 4 subgroups, each subgroup contains two channels of the input data. We can then use 4 kernels $\{\boldsymbol{U}_i\}$ of size $2 \times k \times k$ to convolute with input data and output 4 feature maps. Finally, we subtract the partial features maps from the full feature maps to obtain the temporal feature maps. For the partial feature map$\{\boldsymbol{U}_1\}$, it contains the convolution result of the first 2 channels of the input data. When we subtract it from the full feature map$\{\boldsymbol{W}_1\}$, we can obtain the residual information. The rest of 3 residual feature maps are obtained similarly.

C. SPATIAL CONVOLUTION PACKING (SCP) PART II

Mathematically, let us denote the input feature maps of the l-th convolution layer as $\boldsymbol{I}_l \in \mathbb{R}^{M \times N \times n_{l-1}}$, where M, N and n_{l-1} are the height, width and channel of the input feature maps, and the parameters to be learned include the weights and the biases are indicated as $\theta = (\boldsymbol{W}_l, \boldsymbol{b}_l)$. After the convolution, an activation function $f = \Phi(x)$ maps the convolution result into a nonlinear data space. The 2D convolution process can be described as:

$$f(\boldsymbol{I}_l; \theta_l) = \Phi_l(\boldsymbol{W}_l * f_{l-1}(\boldsymbol{I}_{l-1}; \theta_{l-1}) + \boldsymbol{b}_l) \tag{1}$$

Assume that the dimension of filters at l-th layer is $n_{l-1} \times k_l \times k_l \times n_l$, where n_{l-1} is the number of input feature maps (channel of filters), k_l is the size of filters and n_l is the number of output feature maps. The symbol "$*$" represents the convolution in the spatial domain. As shown in Eq. 2, the 2D convolution does not include the channel domain.

$$I_l^{xy} = \sum_{z=0}^{n_{l-1}} \sum_{h=0}^{k_l-1} \sum_{w=0}^{k_l-1} W_l^{xyz} I_{l-1}^{x+h,y+w,z} + b_l \tag{2}$$

where (x, y) is the position of the l-th output feature map, and is the l-1-th input data at position $(x + h, y + w, z)$ within the receptive field of kernel W_l. During training, each filter convolutes all the input feature maps to output one feature map. That is, for each output feature map, it is the sum of the globally weighted addition of all input feature maps ("2D full conv." in Fig. 2(a)) so that the local correlation among inputs is ignored.

On the other hand, the 3D convolution makes use of filters with dimension $d_{l-1} \times k_l \times k_l \times n_l$, where $d_{l-1} < n_{l-1}$, to calculate:

$$I_l^{xyz} = \sum_{z=0}^{d_{l-1}} \sum_{h=0}^{k_l-1} \sum_{w=0}^{k_l-1} W_{l-1}^{xyz} I_{l-1}^{x+h,y+w,z+t} + b_l \tag{3}$$

where t is the temporal step on channel dimension. This means channel dimension is added to the calculation so that the filter swaps through the 3D space. Hence, each filer covers a cubic region of input feature maps, and each output feature map is the locally weighted addition of input feature maps.

To efficiently extract the temporal correlation across the channel domain, our proposed SCP method replaces the 3D CNN with a much simpler pseudo 3D convolution. As described in Fig. 2, the process is done in three steps: full convolution, partial convolution, and then temporal residual extraction. Full convolution is done the same as the 2D convolution described in Eq. 1. We denote the output of the full 2D convolution as $\boldsymbol{I}_l^S \in \mathbb{R}^{M \times N \times n_{l-1}}$, where $\boldsymbol{S}$ is used to represent the 2D convolution results.

Different from 2D convolution, the partial convolution only uses partial input feature maps to obtain the output feature maps. It means that each output feature map only looks at a subset of input feature maps, which can reduce the number of parameters and introduce local correlation of feature maps for training. The partial convolution is described in Eq. 4:

$$\left\{ \begin{array}{c} I_l^{xyz_g} = \sum\limits_{h=0}^{k_l-1} \sum\limits_{w=0}^{k_l-1} \sum\limits_{z \in z_g} W_{l-1}^{xyz_g} I_{l-1}^{x+h,y+w,z_g} \\ +b_l, \ for \ g = 1, 2, \ldots, G \end{array} \right\} \tag{4}$$

Assuming we pack the input and output feature maps into G groups. The g-th group of input only responds to the g-th group of output I_l^{h,w,z_g}, respectively. Let us denote the partial convolution output as $\boldsymbol{I}_l^P \in \mathbb{R}^{M \times N \times n_{l-1}}$, where $\boldsymbol{P}$ represents the partial convolution. For the final output, a temporal residual $\boldsymbol{I}_l^T$ extraction is obtained by using $\boldsymbol{I}_l^S$ and $\boldsymbol{I}_l^P$ as shown in Eq. 5.

$$\boldsymbol{I}_l^T = \Phi_l(\boldsymbol{I}_l^S) - \boldsymbol{I}_l^P \tag{5}$$

In Eq. 5, the activation function Φ(x) is only used on the output of the full 2D convolution, $\boldsymbol{I}_l^S$, to introduce the

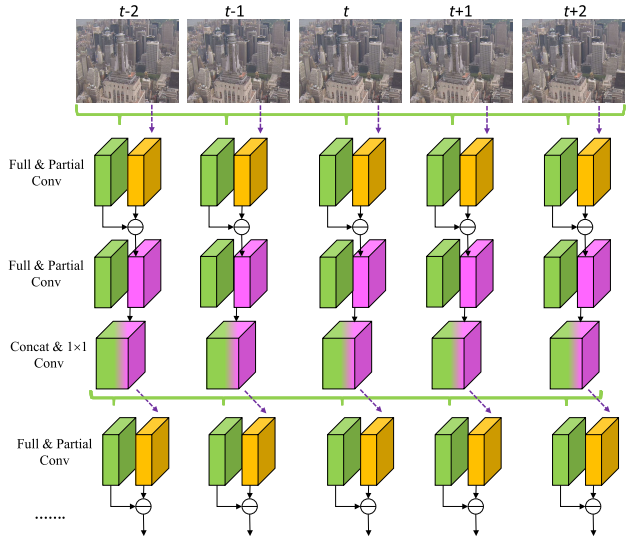

FIGURE 3. SCP scheme for processing video frames.

nonlinearity and the partial convolution, $\boldsymbol{I}_l^{\boldsymbol{P}}$, is directly subtracted from the activated $\boldsymbol{I}_l^{\boldsymbol{S}}$. The activation function can be added to $\boldsymbol{I}_l^{\boldsymbol{S}}$ or $\boldsymbol{I}_l^{\boldsymbol{P}}$ to ensure the nonlinear mapping so that the model can extract nonlinear complex temporal information. Otherwise, the simple sum between $\boldsymbol{I}_l^{\boldsymbol{S}}$ and $\boldsymbol{I}_l^{\boldsymbol{P}}$ only extracts the linear relationship, that cannot learn meaningful features. This design also has some similarity as the residual block in [24] and [28] in the spatial domain. However, our design reduces unnecessary activation layers (the connection of a feature map to a nonlinear function is referred to as a nonlinear mapping) on the shortcut connection and works as a modification of residual learning in the temporal domain rather than the spatial domain. By using the proposed SCP scheme, the spatial and temporal information is jointly extracted and learned for video SR. For visual elaboration, Fig. 3 gives an example of using SCP scheme to process 5 neighborhood frames.

The full convolution (green boxes) extracts the global features from 5 input frames and the partial convolution (yellow boxes) extracts local features from each frame. Their differences (purple boxes) approximate the temporal correlations between each frame to the other 4 frames. After concatenation and another 1×1 convolution, we combine the temporal and full convolution results to form spatial-temporal feature maps to perform another SCP process to extract temporal correlation between each spatial-temporal feature map to the other 4 feature maps. This SCP based convolution can be stacked in multiple layers to form deeper and more complex spatial-temporal feature representation for video SR.

D. EFFICIENT RESIDUAL BLOCKS FOR SPATIAL-TEMPORAL CONVOLUTION NEURAL NETWORKS

Different from image SR, the key for video SR is the long-term feature extraction to model the temporal correlation. Our proposed SCP scheme can be embedded into any image based SR model to replace the regular convolution process for video SR. Inspired by the recent works [25], [26] on their state-of-the-art SR performance, we come up with the ST-CNN model in Fig. 1 that uses back projection based residual blocks to design a deep CNN structure for video SR. As mentioned in Part 3.1, the ST-CNN model has three stages:

Feature Extraction: This is a pre-processing stage. For C input LR frames $\boldsymbol{X}^{LR} \in \mathbb{R}^{H \times W \times C}$, it is important to extract the spatial and temporal correlation as much as possible for further reconstruction. Unlike image SR, the input is usually one single Y-channel or RGB image, ST-CNN takes C frames at once. We use two layers of the proposed SCP based convolution layers to decompose frames into a bigger feature space (first two yellow blocks in Fig. 1). It is equivalent to convolve the frames with a set of filters to represent them in a sparser space. From another perspective, the feature extraction can also be regarded as a sampling process in the spatial and temporal domains using different sampling frequencies. Like 3D Wavelet transformation [49], the sampling process across the spatial and time domains can increase the receptive fields to generate more feature maps.

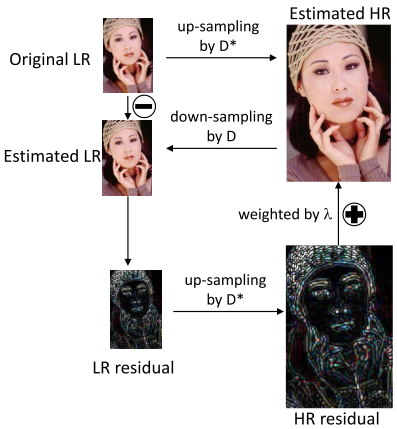

FIGURE 4. Example of back projection process.

Hierarchical Residual Update: The second stage is a crucial part of the ST-CNN. Our modified ST-CNN is modular in back projection structure and each regular convolution is replaced by our proposed SCP based convolution. The basic module is the back projection unit. It includes two up-sampling back projection (UBP) units and a down-sampling back projection (DBP) unit stacked in a hierarchical order. Before introducing the proposed ST-CNN model, let us revise the classic application of back projection in image SR. Its process is shown in Fig. 4. The estimated HR image is updated by calculating the LR residual between the original LR and the down-sampled HR image. For a complete UBP, we can describe the operations mathematically as,

$$I^{HR}(t+1) = I^{HR}(t) - \lambda \cdot H^* D^* \left(DHI^{HR}(t) - I^{LR} \right) \quad (6)$$

where t is the iteration number, H and D are the blurring and down-sampling operators. $H*$ and $D*$ are the inverse operations to work as deblurring and up-sampling operators

and λ is the weighting factor to control the number of residual values for the update. In each iteration, SR image $\boldsymbol{I}^{HR}(t)$ needs to be down-sampled to the same size as the LR image $\boldsymbol{I}^{LR}$, and to calculate the prediction residue. We then use the deblurring and up-sampling operators to up-scale the LR residual back to the desired resolution and add it back for the next iteration. Not only the up-sampling process can be updated by Eq. 6, DBP can also be updated as follows:

$$\boldsymbol{I}^{LR}(t+1) = \boldsymbol{I}^{LR}(t) + \lambda \cdot \boldsymbol{HD}\left(\boldsymbol{D}^{*}\boldsymbol{H}^{*}\boldsymbol{I}^{LR}(t) - \boldsymbol{I}^{HR}\right) \quad (7)$$

In this way, the output of UBP becomes the input of DBP. The above is for image SR which is used in [25]. Our proposed ST-CNN improves the structure by replacing the LR and HR images with their feature maps as follows,

$$\begin{aligned}\boldsymbol{I}_l^{SR} &= \Lambda_l\left(\boldsymbol{I}_l^{SR}\right) - \Phi\left(\boldsymbol{W}_l^{U} * \Phi\left(\boldsymbol{W}_l^{D} * \boldsymbol{I}_l^{SR} - \boldsymbol{I}_{l-1}^{LR}\right)\right)\\ \boldsymbol{I}_l^{LR} &= \Lambda_l\left(\boldsymbol{I}_l^{LR}\right) - \Phi\left(\boldsymbol{W}_l^{D} * \Phi\left(\boldsymbol{W}_l^{U} * \boldsymbol{I}_l^{LR} - \boldsymbol{I}_{l-1}^{SR}\right)\right)\end{aligned} \quad (8)$$

where Λ is the 1×1 weighting convolution, and $\boldsymbol{I}_l^{SR}$ is the output of UBP and $\boldsymbol{I}_l^{LR}$ is the output of DBP at l-th layer. To illustrate the details of our proposed residual block, we compare the residual blocks among ResNet [28], DBPN module and our modified ST-CNN module in Fig. 5.

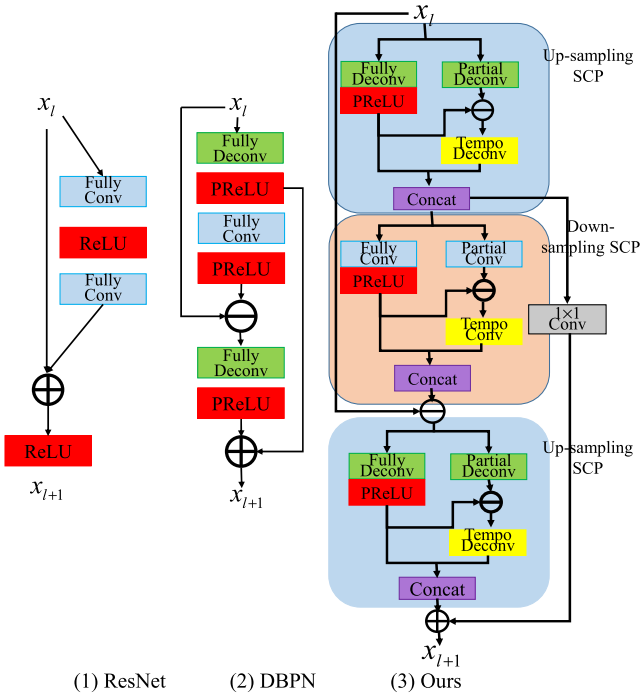

FIGURE 5. Residual blocks comparison among ResNet, DBPN and our approach.

In Fig. 5(3), the blue block represents the convolution process that output LR feature maps, the green block represents the deconvolution process that outputs HR feature maps, the yellow block represents the partial convolution process on the temporal domain and the gray block represents the 1×1 convolution that works as weighting process. Compared to the residual block in ResNet, our proposed residual block has the same back projection process as the residual block in DBPN that contains up-sampling and down-sampling processes. Each block includes two shortcuts between the HR and LR features to update the prediction residuals. The major improvement of the residual blocks between the proposed ST-CNN and DBPN include the following. 1. Embedding the SCP process at each convolution/ deconvolution layer to extract the temporal correlation (the up-sampling and down-sampling SCP processes as shown in Fig. 4(3)): full convolution is followed by an activation layer to introduce the nonlinear mapping, and then we concatenate the full convolution and extract temporal results together as input for the next layer. 2. Adding an extra 1×1 convolution to form weights on the predicted HR residual on updating the HR feature maps (the gray block in Fig. 5(3)): for the residual block in ResNet, its task is to train a general image classifier so it does not care for pixels differences. Simple shortcut can help to avoid gradient vanishing when the model gets deeper. In video SR, it is important to keep the fidelity of pixel based reconstruction. As described in Eq. 6, the residual image for back projection can be controlled by the weighting factor which in our modified residual block, 1×1 convolution is used to simulate the weighting factor which acts as an updating rate to avoid sudden change. It can also be considered as an adaptive regression model that assigns weighting values for minimizing the variance of the distribution of residuals.

Furthermore, to make full use of the residual of LR and HR feature maps and to avoid gradient vanishing, skip connection is used between UBP and DBP. There is a concatenation layer to aggregate the previous outputs as input for the next unit. For the l-th UBP (Eq. 9a), the input is a combination of $\{\boldsymbol{I}_{l-1}^{LR}, \boldsymbol{I}_l^{LR}\}$ and for l-th DBP (Eq. 9b), the input is a combination of $\{\boldsymbol{I}_{l-1}^{HR}, \boldsymbol{I}_l^{HR}\}$. We call this global connection because it connects the intermediate outputs from the previous back projection unit. Moreover, we also propose a local connection within each back projection unit. For the UBP or DBP unit, the previous weighted output can also be shared across different units to reuse the extracted features. We then have the description of local and global connections as follows:

$$\begin{aligned}\boldsymbol{I}_l^{HR} = {} & \underbrace{\boldsymbol{\Lambda}_l(\boldsymbol{I}_l^{HR}) + \boldsymbol{\Lambda}_{l-1}(\boldsymbol{I}_{l-1}^{HR})}_{local}\\ & -\boldsymbol{\Phi}\left(\boldsymbol{W}_l^{U} * \boldsymbol{\Phi}\left(\boldsymbol{W}_l^{D} * \boldsymbol{I}_l^{HR} - \underbrace{\{\boldsymbol{I}_{l-1}^{LR}, \boldsymbol{I}_l^{LR}\}}_{global}\right)\right)\end{aligned} \quad (9a)$$

$$\begin{aligned}\boldsymbol{I}_{l+1}^{LR} = {} & \underbrace{\boldsymbol{\Lambda}_{l+1}(\boldsymbol{I}_{l+1}^{LR}) + \boldsymbol{\Lambda}_l(\boldsymbol{I}_{l-1}^{LR})}_{local}\\ & -\boldsymbol{\Phi}\left(\boldsymbol{W}_{l+1}^{D} * \boldsymbol{\Phi}\left(\boldsymbol{W}_{l+1}^{U} * \boldsymbol{I}_{l+1}^{LR} - \underbrace{\{\boldsymbol{I}_{l-1}^{HR}, \boldsymbol{I}_l^{HR}\}}_{global}\right)\right)\end{aligned} \quad (9b)$$

In Fig. 6, we show the connections between UBP and DBP. There are four units in the order of UBP-DBP-UBP-DBP.

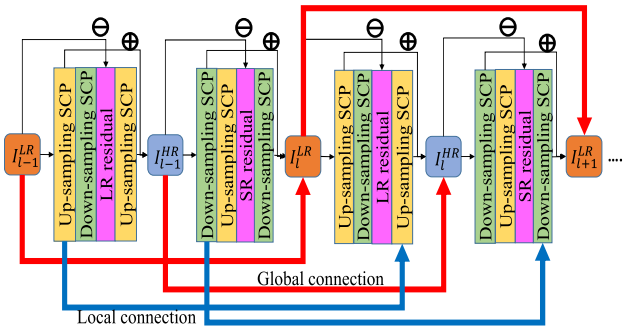

FIGURE 6. Global and local connection in ST-CNN model.

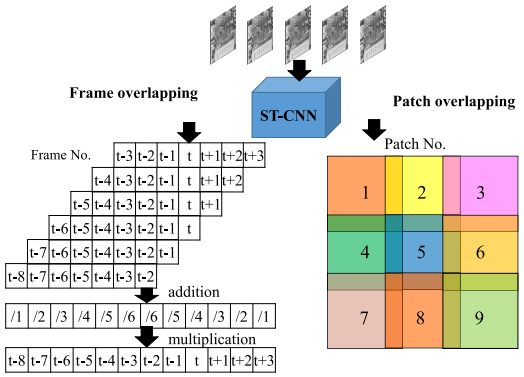

FIGURE 7. Frame and patch overlapping for video SR enhancement.

The global connections (red lines) are to enable the sharing of the input features with different units. This global connection connects the input and output of different units. The local connections (blue lines) reuse the intermediate convolutional results for computation. We call it a local connection because it makes use of the local features across different units.

Final reconstruction. The last stage is to aggregate all intermedia results together for generating the final SR frames. Instead of outputting one SR frame from C (number of frames) LR frames, ST-CNN outputs the same number of SR frames as the number of input LR frames. This means that for the same video sequence with U frames, it only needs U/C forward computation. Instead of using SCP convolution, we use two full convolution layers to concatenate all the outputs of the back projection units together and use the Mean Absolute Error (MAE) [24] to replace the Mean Square Error (MSE) to calculate the SR loss as follows:

$$loss_{MAE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left\|\boldsymbol{Y}_{i,j}^{HR} - \boldsymbol{I}_{i,j}^{SR}\right\|_1 \qquad (10)$$

where N is the number of batches, $\boldsymbol{Y}_{i,j}^{HR}$ is the ground truth HR frame and $\boldsymbol{I}_{i,j}^{SR}$ is the predicted SR frame.

E. SELF-ENSEMBLE ENHANCEMENT IN TEMPORAL DOMAIN

Through the three stages of convolution, our ST-CNN model can generate C SR frames at once. It is helpful for fast implementation but when encountering videos with dynamic motions or complicated patterns, this coarse SR can be less accurate. To improve the SR quality, two simple approaches can be used: frame overlapping and patch overlapping.

For frame overlapping, let us denote it as ST-CNN(F+). We determine the frame step of video SR as T_f ($T_f \leq C$). For super-resolving challenging videos, T_f can be smaller than the number of LR frames to overlap output SR frames. The process is shown in Fig. 7.

Similarly, patch overlapping is also shown in Fig. 7. It is used when the memory of GPU is not enough. Let us denote it as ST-CNN(P+). The LR frames need to be split into overlapping patches and perform patch-based video SR by patch step T_P. This patch-based SR has been proven to be effective in image SR in [25], [26]. In our experiments, we will show that both frame overlapping and patch overlapping is very useful for video SR.

IV. EXPERIMENTAL RESULTS

To verify the accuracy and efficiency of our proposed video SR methods, we applied our model to different video datasets, and present quantitative and qualitative results for comparison.

A. DATASETS AND IMPLEMENTATION DETAILS

For the training process, we used the standard video sequences[1] the same as in [34]–[39]. For HR frames, based on the up-sampling factor α, we used bicubic from MATLAB to down-sample HR frames α time to obtain LR frames. For both LR and HR frames, we converted them into YUV color space and used Y-channel data only because the Y-channel contains enough information for video processing while U and V channels contain color information that can simply be up-sampled by Bicubic. The input LR patch was cropped from the LR frames with size $32\times32\times C$, where C is the number of adjacent frames. The HR patch has the size of $32\alpha\times32\alpha\times C$. To enlarge the training data, we cropped multiple overlapped volumes (groups of adjacent frames) from training videos. Eventually, we can generate around 400,000 training LR-HR data pairs.

We tested our model on Set2[2] (Vid4: calendar, city, foliage and walk) with recent state-of-the-art video SR methods: VSRnet [36] (Video Super-Resolution with Convolutional Neural Networks, first CNN based video SR), Bayesian [34] (Bayesian based learning approach), VESPCN [37] (pixel alignment network), BRCN [40] (3D CNN network), SAN [39] (spatial alignment with optical flow network) DF [38] (sub-pixel motion compensation network) and TOFlow [51] (Optical flow compensation for joint video denoising, deblurring and super-resolution). Meanwhile, we also tested our ST-CNN network on 4K videos[3] to show the visual quality. This dataset contains seven

[1] 1https://media.xiph.org/video/derf/
[2] 2https://people.csail.mit.edu/celiu/CVPR2011/
[3] 3http://ultravideo.cs.tut.fi/

120fps sequences: *Beauty, Bosphorus, HoneyBee, Jockey, ReadySteadyGo, ShakeNDry* and *YachtRide*.

B. NETWORK ARCHITECTURE

Our ST-CNN model uses SCP based convolution for the first and second stages and uses the common full 2D convolution at the third stage. For video SR with up-sampling factor $\alpha = 3$, we used convolution/deconvolution layers with 7×7 filters, three striding and four padding. As for up-sampling factor $\alpha = 4$, we used convolution/deconvolution layers with 8×8 filter, four striding and four padding. To achieve fast video SR, we reduced the number of filters at deconvolution layers to 20 for up-sampling and 40 for convolution layers for down-sampling. For other feature extraction layers and concatenation layers, we used filters of size 3×3. The design of these filters is based on general intuition of preliminary experiments to tradeoff between accuracy and efficiency.

We initialized the weights based on [27] and all convolution layers are followed by parametric rectified linear units (PReLUs). We trained our model with learning rate initialized to 0.0001 for all layers and decreased by 10 after 800,000 for a total of 1,000,000 iterations. For optimization, we used Adam with momentum equals 0.9 and weight decay equals 0.0001. All experiments were conducted using Caffe, MATLAB R2016b on two NVIDIA GTX 1080 Ti GPUs.

C. ANALYSIS OF NETWORK STRUCTURE

In this section, let us discuss important parameters valuable to achieve good structures and give some further analysis.

1) EFFECT OF THE SCP SCHEME

To prove the efficiency of the Spatial Convolution Packing (SCP) scheme, we ported it into other state-of-the-art CNN models by replacing the conventional 2D convolution layers for video SR, including VDSR [18], VSRnet [36], VESPCN [37] and DBPN [25]. More specifically, for one 2D convolution layer with N filters in these CNN models, we replaced them by $N/2$ filters for full convolution and $N/2$ filters for partial convolution to ensure that the number of filters remains the same for a fair comparison. We have denoted the approaches using SCP based convolution as SCP models. We tested the performance on Set2 video with an up-sampling factor of 4 and then obtained TABLE 1. We have named the modified versions as VDSR-SCP, VSRnet-SCP and VESPCN-SCP for clarity. For VDSR, VSRnet, VESPCN and DBPN, they were originally proposed for single image SR. To modify them to VDSR-SCP, VSRnet-SCP, VESPCN-SCP and DBPN-SCP, we consider each color channel as an individual image so that RGB color images can be replaced as 3 neighborhood frames. Hence, we can adopt the models to perform SCP based convolution. As for the proposed ST-CNN, we also tested it without SCP and labelled it as ST-CNN(−ve).

Table 1 shows the experimental results in PSNR. We can see that using SCP always achieves better SR performance in terms of PSNR and reduces the running times. It shows that using the SCP can effectively extract the temporal information for video SR. Because of replacing half of the full convolution with the partial convolution, it is the reason for saving computation time compared to the original versions. Note also that VSRnet has 3 convolution layers while VDSR, VESPCN and DBPN can be modeled up to 20 to 40 convolution layers. The improvement on PSNR also proves that using a deeper CNN model, to some extent, can help to boost up the performance of video SR.

TABLE 1. Results of PSNR (dB) and time (sec) by comparing CNN models with and without SCP scheme on set2 of 4× SR.

Model	Layer no.	SCP	PSNR	SSIM	Time
VSRnet	3		22.79	0.5968	0.65
VSRnet-SCP	3	√	22.90	0.6101	0.56
VDSR	20		24.73	0.6321	0.77
VDSR-SCP	20	√	24.85	0.6330	0.71
VESPCN	38		25.33	0.6742	0.76
VESPCN-SCP	38	√	25.41	0.6750	0.70
DBPN	42		25.49	0.7012	0.88
DBPN-SCP	42	√	25.60	0.7145	0.87
ST-CNN(-ve)	45		25.85	0.7889	0.34
ST-CNN	45	√	**25.96**	**0.8001**	**0.24**

2) EFFECT OF GLOBAL AND LOCAL CONNECTIONS FOR ST-CNN MODEL

The main structure of our proposed STCNN model is built based upon the back projection based residual blocks. The main improvement compared to [25] and [26] comes from global and local connections which allow to share the residual feature maps for reconstruction. Similarly, we tested the ST-CNN model with and without global and local connection on Set2 with an up-sampling factor of $\alpha = 4$.

TABLE 2. PSNR (dB) and time (sec) comparison of the ST-CNN model with and without global and local connections for video SR on set2 of 4× SR.

model	Global connection	Local connection	PSNR	SSIM	Time
A			25.79	0.762	**0.20**
B	√		25.87	0.765	0.22
C		√	25.89	0.765	0.22
D	√	√	**25.96**	**0.800**	0.24

Table 2 shows the comparison of video SR performance on ST-CNN model with and without global and local connection (Noted that the ST-CNN used SCP scheme for spatial-temporal convolution). For clarity, we name different ST-CNN models as A, B, C and D to identify the use of global and local connections as shown in TABLE 2. For PSNR results of the SR, we can see that using global and local connections (B, C, D in TABLE 2) can improve the SR quality about 0.10∼0.17 dB. For the computation times, we can see no significant increasing because the

local and global connections use 1×1 convolution layer as a weighting matrix. When combining both global and local connections (D in TABLE 2), it not only assists the residual information to be used across the LR and HR feature maps, but also alleviates the vanishing gradient problem to produce improved features.

3) EFFECT OF NUMBER OF FRAMES AS INPUT FOR ST-CNN TRAINING

In the proposed ST-CNN model, we input C LR frames and output C SR frames to speed up the SR process. The value of C affects how accurate and efficient video SR results we can be achieved. To find the optimal value of C, we tested ST-CNN model on Set2 videos with different motions for comparison and with up-sampling factor $\alpha = 4$. The results are shown in Table 3.

TABLE 3. The PSNR (dB) and time (sec) by comparing ST-CNN model with different value of C on set2 of 4× SR.

C	PSNR	SSIM	Time
3	25.78	0.762	0.43
4	25.84	0.764	0.36
5	**25.96**	**0.800**	0.24
6	25.68	0.758	0.21
10	25.44	0.754	**0.13**

From the results, there is a tradeoff between PSNR and running time. Generally, the more adjacent frames we super-resolve at once; the faster implementation we can get because for the same video sequence with U frames, ST-CNN needs U/C times of computation, hence the larger C means fewer iterations. However, as shown in Eq. (10), the more LR frames we have to up-sample, the more missing pixels of HR frames need to be predicted. The MAE calculation is an average squared pixel differences between C SR frames and HR frames. When C becomes too large, the model cannot guarantee the quality for each SR frame. Meanwhile, larger C means longer-term temporal dependency which may require deeper and more complex networks for modelling. Considering the tradeoff between good SR quality and fast running time, $C = 5$ is a good choice which gives the highest PSNR and is the third fastest computation.

D. COMPARISON WITH STATE-OF-THE-ART METHODS

Let us select network architectures with the best performance to compare our default ST-CNN, including:

- Bicubic: bicubic interpolation
- VSRnet [36]: first CNN based video SR with shallow CNN structure.
- Bayesian [34]: Bayesian approach with fast running time.
- VESPCN [37]: sub-pixel alignment network modified from image SR with good performance.
- BRCN [44]: 3D model using bidirectional recurrent structure.

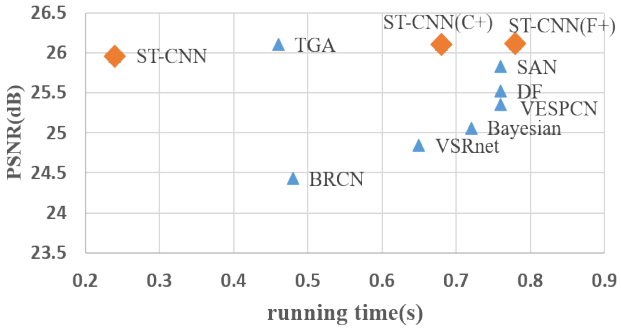

FIGURE 8. PSNR vs time comparison on proposed methods with other state-of-the-art methods onSet2 of 4× SR.

- DBPN [25]: state-of-the-art image SR approach
- SAN [39]: combination of spatial alignment and optical flow together to train a network for video SR.
- DF [38]: sub-pixel motion compensation network for better detail reveal.
- TOFlow [51]: Optical flow compensation for joint video denosing, deblurring and super-resolution.
- TGA [57]: Video SR with Temporal Group Attention

For all these methods, we used their publicly available codes or SR results from their respective publications. Among them, the authors of VESPCN, BRCN, SAN and DF just recently announced their results without any available codes for comparison. We used the results on Set2 from their papers for comparison. All results are shown in TABLE 4 with up-sampling factors 3 and 4, in terms of PSNR (dB), SSIM, NIQE [59], FLOPs (TMAC) and running time (s). PSNR and SSIM are used to quantitatively evaluate data fidelity of SR. Higher values indicate better reconstruction. NIQE is a non-reference metric used for qualitative evaluation, and lower values indicate better visual quality. FLOPs [60] were calculated on Set2 dataset and running times were obtained by running different methods on Set2 dataset using the same machine.

From Table 4, we can see that our proposed methods can achieve both the best running times and the highest PSNR compared with state-of-the-art methods. It can be found that TGA is one that is close to our approach on PSNR and FLOPs. However, it costs more running time because it requires global softmax normalization on the feature map which can be time-consuming. On average, our ST-CNN model can improve the SR quality by about 0.13∼3.00 dB in terms of PSNR, while FLOPs are reduced by about 0.5∼0.7 TMAC and the running time is reduced by about 0.3∼0.4 seconds. For NIQE, our proposed method can outperform others by about 0.1∼0.2. For ST-CNN(F+), we overlapped the output SR frames by frame step $T_f = 2$ to achieve better PSNR performance. For ST-CNN(P+), we extracted 120×120 patches from the LR frame as input and overlapped the patch by patch step by Tp $= 32$. Because of the overlapping of frames and patches, ST-CNN(F+) and ST-CNN(P+) can generate smoother and more consistent spatial and temporal changes, which shows the significant use of self-ensemble

TABLE 4. Comparison of the proposed ST-CNN and other video SRs on Set2 dataset in terms of PSNR (dB), SSIM, NIQE, FLOPs (TMAC) and running time (sec).

Method	Est.	Bicubic	VSRnet [36]	Bayesian [34]	VESPCN [37]	BRCN [44]	DBPN [16]	SAN [39]	DF [38]	TOFlow [51]	TGA [57]	ST-CNN (Proposed)	ST-CNN (F+) (Proposed)	ST-CNN (P+) (Proposed)
×3	PSNR	25.29	25.31	25.64	27.25	-	-	-	27.49	-	-	**27.91**	**28.14**	**28.10**
	SSIM	0.7325	0.7648	0.8015	0.8426	-	-	-	0.8431	-	-	**0.8702**	**0.8845**	**0.8814**
	NIQE	4.841	4.778	3.762	3.550	-	-	-	3.201	-	-	**2.787**	**2.745**	**2.745**
	FLOPs	-	0.0001	-	0.001	-	-	-	0.42	-	-	**0.03**	**0.15**	**0.24**
	Time	0.001	0.76	0.80	0.84	-	-	-	0.84	-	-	**0.23**	**0.72**	**0.66**
×4	PSNR	23.80	24.84	24.66	25.35	24.43	25.37	25.83	25.52	25.89	26.10	**25.96**	**26.12**	**26.10**
	SSIM	0.6325	0.6586	0.7425	0.6848	0.6334	0.737	0.7668	0.7600	0.765	0.8254	**0.8001**	**0.8230**	**0.8113**
	NIQE	5.113	5.001	3.997	3.695	3.703	3.590	3.421	3.339	3.292	3.010	**3.118**	**3.008**	**3.008**
	FLOPs	-	0.0001	-	0.004	12.1	9.90	0.55	0.62	0.81	0.07	**0.06**	**0.30**	**0.48**
	Time	0.001	0.65	0.72	0.76	0.48	0.88	0.767	0.76	0.95	0.46	**0.24**	**0.78**	**0.68**

enhancement. The setting was determined by experiments. The value of the step can also be modified based on videos with different features and motions. To better visualize the tradeoff between PSNR and running time, Fig. 8 gives a diagram to show the performance of all methods. The left upper corner represents the higher PSNR with less computation time. Our proposed methods (ST-CNN, ST-CNN(F+), ST-CNN(P+)) can outperform other state-of-the-art methods in PSNR or running time.

TABLE 5. The PSNR (dB) and time (sec) comparison between SAN and our proposed methods on 4K videos by 4× up-sampling.

Video	SAN	ST-CNN	ST-CNN(F+)
Beauty	35.95	39.80	40.11
	(0.80)	(0.31)	(0.65)
Bosphorus	43.53	40.80	43.68
	(0.96)	(0.38)	(0.61)
HoneyBee	40.02	41.31	40.21
	(0.91)	(0.33)	(0.76)
Jockey	41.21	42.41	41.28
	(0.93)	(0.35)	(0.81)
ReadySteadyGo	41.17	36.98	41.27
	(0.96)	(0.39)	(0.87)
ShakeNDry	39.70	38.09	39.92
	(0.92)	(0.34)	(0.80)
YachtRide	38.03	35.14	38.31
	(0.95)	(0.37)	(0.86)
Average	39.94	39.21	40.13
	(0.92)	(0.35)	(0.76)

E. COMPARISON ON REAL-WORLD VIDEOS

In Set2 dataset, the video is of 720p format which becomes less popular nowadays. To measure the practical performance of our proposed methods on real-world videos, we did two experiments: 1) 4× super-resolution tests on 4K videos. This video dataset consists 7 videos in 2160p format. *ShakeNDry* has 300 frames while the others have 600 frames. Since SAN has proven to have good performance for Set2 dataset, we only make a comparison with them. TABLE 5 shows a comparison of the results on each video sequence. 2) Blind video SR. We used the video clip from CamSeq01,[4] which is a sequence depicting a moving driving scene for object recognition. We directly super-resolved the sequence without knowing any priors. Fig. 10 shows the visualization results.

From Table 5, our proposed ST-CNN uses 0.37 seconds less than that of SAN to achieve similar PSNR performance. For videos with simple features or static motions, like Beauty, HoneyBee and Jockey, the improvement of using ST-CNN is obvious, while other videos with more dynamic motions and complex features (*Bosphorus, ShakeNDry, YachtRide*) can suffer from the many-to-many mechanism of ST-CNN. We further proposed the ST-CNN(F+), which overlaps the SR frames to compensate for the error in adjacent frames, which gives good PSNR improvement. From TABLE 5, the PSNR is improved significantly compared to ST-CNN and it also outperforms SAN about 0.2 dB and reduces 0.16 s in computation time.

F. ANALYSIS OF VIDEO SR ON VISUAL QUALITY

Besides quantitative comparison, we need to visualize the SR results to appreciate the visual quality. We also tried to visualize the intermediate feature maps and trained filters to analyze the representative ability of each convolution layer. For better observation, we suggest readers to view the electronic version of the figures for comparison.

1) VISUAL COMPARISON AMONG STATE-OF-THE-ART SR METHOD

Generally, 4× video up-sampling is more useful for applications in real-time. In this section, we will show some visual comparisons among our approaches and other SR approaches as shown in Figs. 9 and 10. To show the visible differences among different SR methods, we tested 4× up-sampling on Set2. We used the results from VSRnet, Bayesian, DF and SAN to make comparison, and please note that the authors of BRCN have not released their codes and results yet.

[4] 4http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamSeq01

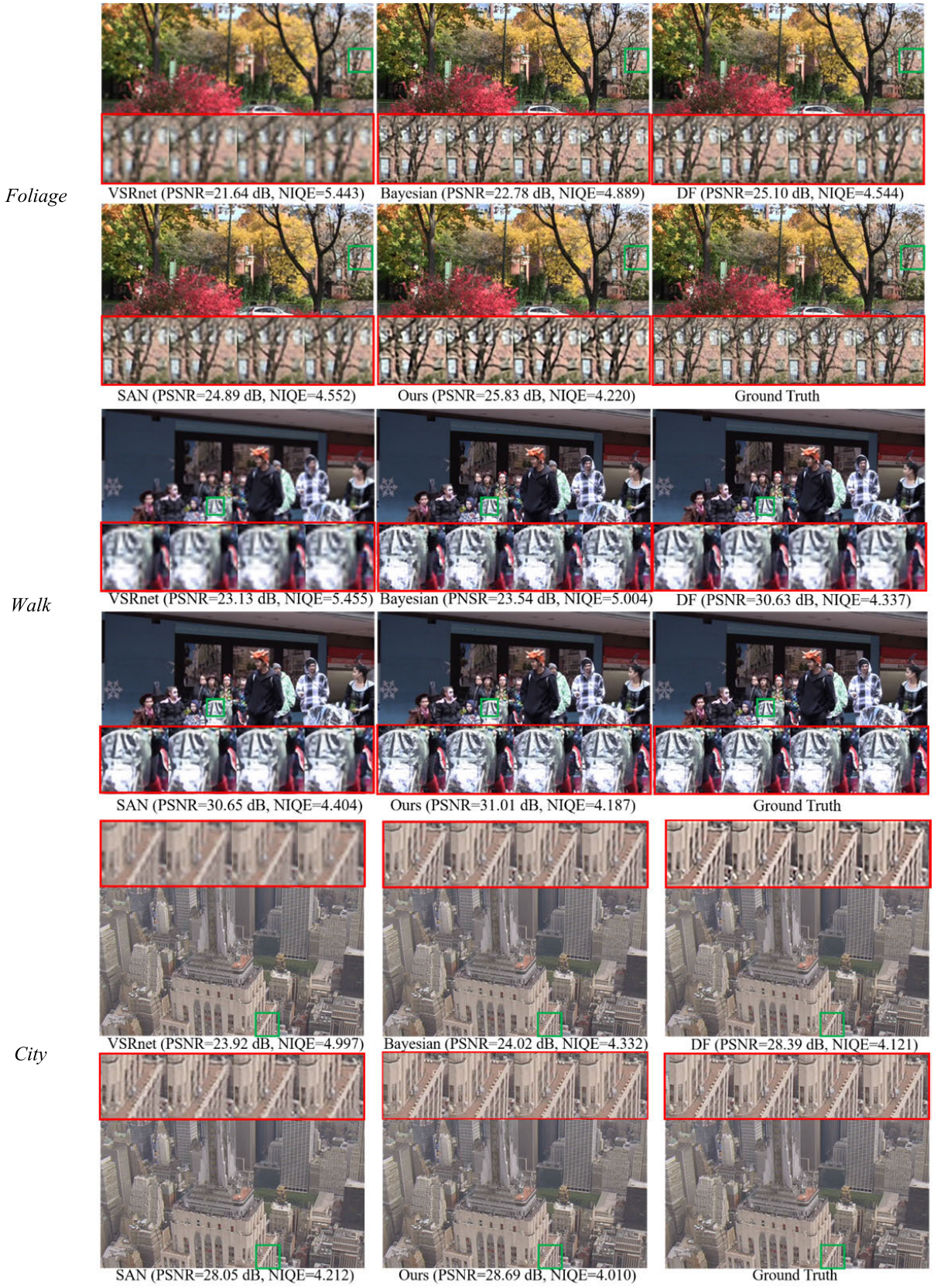

FIGURE 9. 4× up-sampling visual comparison among different video SR approaches in Set2.

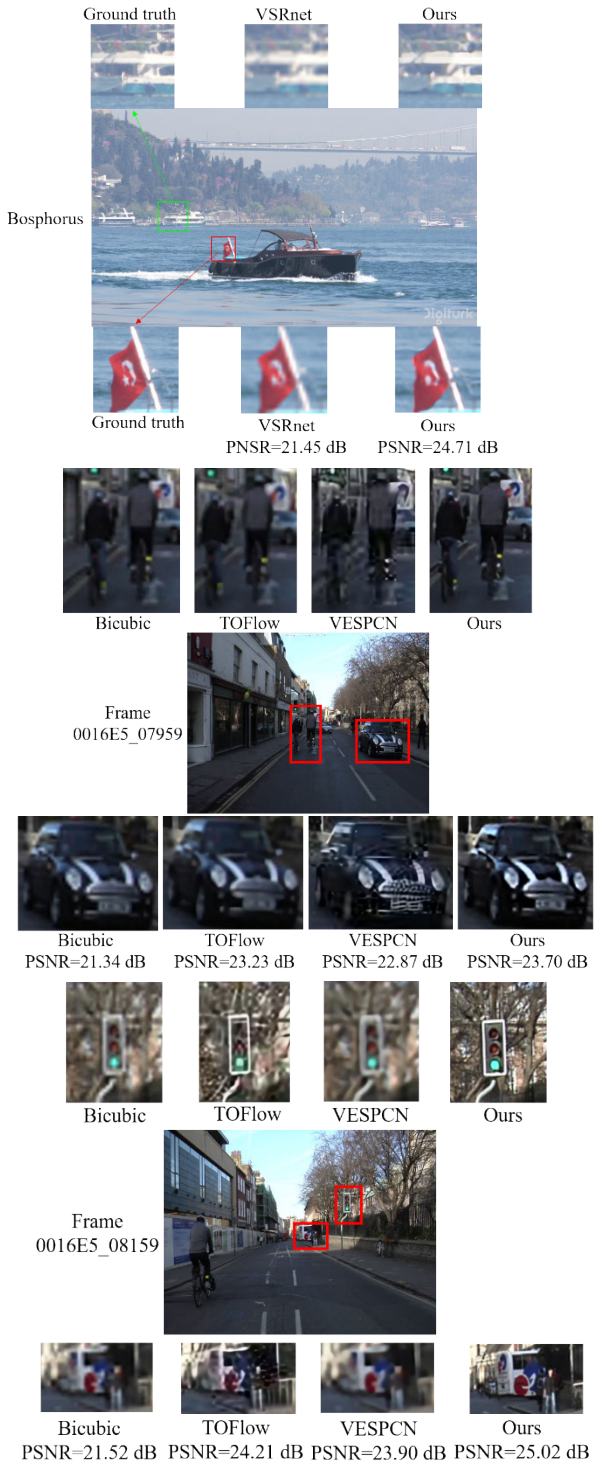

FIGURE 10. 4× up-sampling for 4K video SR and blind video SR.

Three video sequences (*Foliage*, *Walk* and *City*) are listed to show the detailed results of SR. For each SR frame, the same cropped regions (see the green windows on frames) of the adjacent 4 frames are shown to indicate the SR performance relating to the effect on motion changes.

On the *Foliage* image, we can see that the proposed ST-CNN can achieve sharper and clearer tree branches compared to other video SR approaches. As for the *Walk* and *City* images, using our proposed method can also generate much clearer details around the cradle and tower as compared with other SR approaches.

Meanwhile, we also tested the SR performance on 4K video sets and the real footage depicting the driving screen. The results are shown in Fig. 10. Note that the test was also done for 4× up-sampling. *Bosphorus* is one example from the 4K video set. It is seen that our approach can enhance the edge and texture regions (flag and ship) with pleasing quality. Frame *0016E5_07959* and Frame *0016E5_08159* are two examples from the CamSeq01 dataset. Without knowing the 4× ground truth, we only show the visual comparison among different approaches. We can see that our proposed approach can achieve smoother and sharper patterns, like the traffic light, car, bus, and pedestrian.

2) MOTION ANALYSIS OF VIDEO SR

For video SR, it is important to ensure that the super-resolved SR frames contain consistent motions to provide smooth and pleasing visual experiences.

To estimate the motion accuracy, we choose two criteria: frame residues and optical flow loss for illustration. We selected the Walk sequence from Vid4 test set to make the evaluation. For the term "frame residues", it means that we calculate the Euclidean distance between two adjacent frames to estimate the average pixel loss. Larger losses indicate larger differences between adjacent frames. Hence it can roughly calculate the degree of motion changes. Only using frame residues cannot fully show the motion smoothness. Hence, we also used the optical flow field as another measurement. Optical flow tries to calculate the motion difference between two frames which are taken at times t and $t + \Delta t$ for every pixel position. It is a 2D vector showing the displacements of the pixel from frame t to $t + \Delta t$. We used the public package provided by [50] to estimate the optical flow. Then we calculated the Endpoint error (Euclidean distance between HR and SR optical flow) to measure the optical flow loss. The frame residues and optical flow loss are shown in Table 6.

TABLE 6. Comparison among different video SR algorithms on motion estimation.

Methods	Bayesian	BRCN	SAN	DF	ST-CNN
Frame residues	545.14	489.51	298.92	287.68	269.35
Endpoint error	0.6697	0.5513	0.3315	0.2843	0.1986

From Table 6, we can find that our proposed ST-CNN gives the lowest frame residues and Endpoint errors compared with other video SR algorithms. It shows that ST-CNN can generate SR frames with motions close to the ground truth. To better demonstrate the motion compensation, we also visualize the optical flow results by magnitude values in Fig. 11. Ground truth means that the optical flow was generated from the HR frames. Your attention is drawn on the visual differences marked in black circles. Combining the

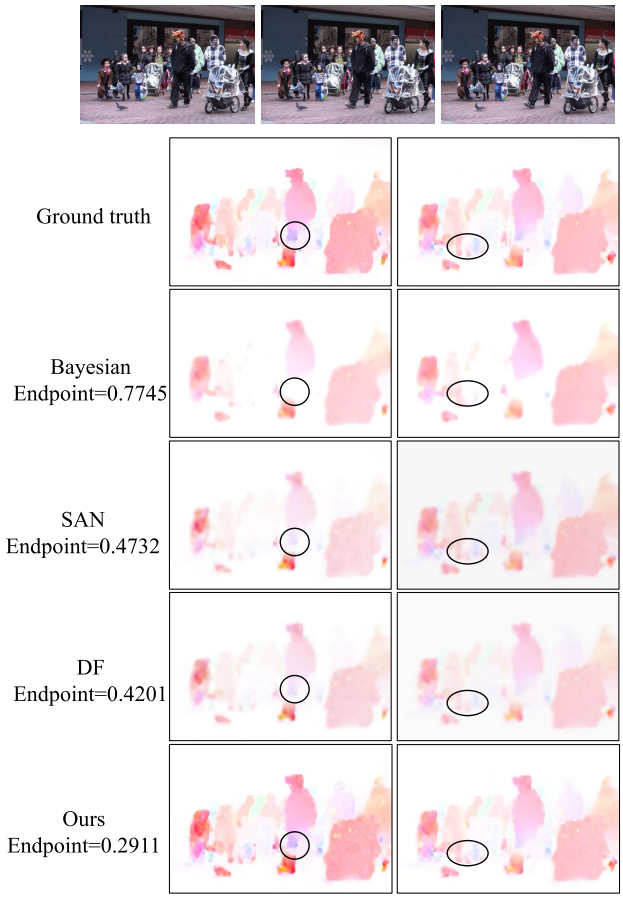

FIGURE 11. Visualization of optical flow.

results in TABLE 6 and Fig. 11, we can see that using the proposed ST-CNN can predict similar or more smooth optical flow compared to other video SR algorithms. It indicates that using the proposed ST-CNN can reconstruct the SR frames with accurate motion compensation.

V. CONCLUSION

In this paper, we propose a spatial-temporal convolution based CNN model for video SR. By using the Spatial Convolution Packing, our proposed ST-CNN model can jointly extract both spatial and temporal features to achieve good SR quality. Compared to the conventional 3D convolution process, ST-CNN can avoid complex calculations and achieve fast implementation. Meanwhile, we optimize the back projection and residual learning blocks to exploit deeper and meaningful feature maps while using fewer filter coefficients. Experimental results on various video datasets show the superior performance of the ST-CNN and its enhanced versions, quantitatively and qualitatively. Results of testing 4K videos also prove that our approach can handle various edge patterns and motions in real-time. For analysis, we have visualized critically the important parameters to demonstrate the usefulness of each convolution layer in ST-CNN. It is demanding to achieve real-time video SR for videos with different resolutions or motions. Executive Codes are available at [61].

REFERENCES

[1] M. Irani and S. Peleg, "Improving resolution by image registration," *Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991.
[2] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image superresolution via sparse representation," *IEEE Trans. Image Process*, vol. 19, no. 11, pp. 2861–2873, May 2010.
[3] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1920–1927.
[4] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. IEEE Int. Conf. Asian Conf. Comput. Vis. (ACCV)*, Singapore: Springer, Nov. 2014, pp. 111–126.
[5] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, Jun. 2011, pp. 449–456.
[6] L. Sun, S. Cho, J. Wang, and J. Hays, "Good image priors for non-blind deconvolution," in *Proc. IEEE Int. Conf. Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland, Sep. 2014, pp. 231–246.
[7] Z.-S. Liu, W.-C. Siu, and J.-J. Huang, "Image super-resolution via weighted random forest," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Toronto, ON, Canada, Mar. 2017, pp. 1019–1023.
[8] L. Zhi-Song and W.-C. Siu, "Cascaded random forests for fast image super-resolution," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 2531–2535.
[9] C. Dong, C. Change Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
[10] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vegas, NV, USA, Jun. 2016, pp. 1646–1654.
[11] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1637–1645.
[12] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1874–1883.
[13] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 624–632.
[14] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4681–4690.
[15] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 136–144.
[16] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 1664–1673.
[17] Z.-S. Liu, L.-W. Wang, C.-T. Li, and W.-C. Siu, "Hierarchical back projection network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2041–2050.
[18] K. Jiang, Z. Wang, P. Yi, G. Wang, K. Gu, and J. Jiang, "ATMFN: Adaptive-threshold-based multi-model fusion network for compressed face hallucination," *IEEE Trans. Multimedia*, vol. 22, no. 10, pp. 2734–2747, Oct. 2020.
[19] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, and J. Cheng, "ODE-inspired network design for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1732–1741.
[20] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 5689–5698.
[21] Z.-S. Liu, L.-W. Wang, C.-T. Li, W.-C. Siu, and Y.-L. Chan, "Image super-resolution via attention based back projection networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3517–3525.

[22] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 2472–2481.

[23] D. Zhang, J. Shao, and H. T. Shen, "Kernel attention network for single image super-resolution," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 3, pp. 1–15, Sep. 2020.

[24] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, and H. Shen, "Single image super-resolution via a holistic attention network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 191–207.

[25] W. Sun and Z. Chen, "Learned image downscaling for upscaling using content adaptive resampler," *IEEE Trans. Image Process.*, vol. 29, pp. 4027–4040, 2020.

[26] S. Anwar and N. Barnes, "Densely residual Laplacian super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 2, 2020, doi: 10.1109/TPAMI.2020.3021088.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1026–1034.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[29] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, Feb. 1994.

[30] Y.-Z. Zhang, W.-C. Siu, Z.-S. Liu, and N.-F. Law, "Learning via decision trees approach for video super-resolution," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Las Vegas, NV, USA, Dec. 2017, pp. 558–562.

[31] Z. Ashouri and S. Shirani, "Video super resolution using contourlet transform and bilateral total variation filter," *IEEE Trans. Consum. Electron.*, vol. 59, no. 3, pp. 604–609, Aug. 2013.

[32] J. Chen, J. Nunez-Yanez, and A. Achim, "Video super-resolution using generalized Gaussian Markov random fields," *IEEE Signal Process. Lett.*, vol. 19, no. 2, pp. 63–66, Feb. 2012.

[33] C. Liu and D. Sun, "On Bayesian adaptive video super resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 346–360, Feb. 2014.

[34] J. Chen, J. L. Nunez-Yanez, and A. Achim, "Bayesian video super-resolution with heavy-tailed prior models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 905–914, Jun. 2014.

[35] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 531–539.

[36] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Trans. Comput. Imag.*, vol. 2, no. 2, pp. 109–122, Jun. 2016.

[37] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jun. 2017, pp. 2848–2857.

[38] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 4482–4490.

[39] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, X. Wang, and T. S. Huang, "Learning temporal dynamics for video super-resolution: A deep learning approach," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3432–3445, Jul. 2018.

[40] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[41] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4489–4497.

[42] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5534–5542.

[43] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 1, pp. 314–323, Jan. 2019, doi: 10.1109/JBHI.2018.2808281.

[44] Y. Huang, W. Wang, and L. Wang, "Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1015–1028, Apr. 2018.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[47] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1106–1114.

[48] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. IEEE Int. Conf. Eur. Conf. Comput. Vis. (ECCV)*, vol. 8689. Amsterdam, The Netherlands: Springer, Oct. 2014, pp. 818–833.

[49] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "Memory-constrained 3D wavelet transform for video coding without boundary effects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 812–818, Sep. 2002.

[50] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[51] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[52] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3897–3906.

[53] M. Haris, G. Shakhnarovich, and N. Ukita, "Space-time-aware multi-resolution video enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2859–2868.

[54] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, "EDVR: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Long Beach, CA, USA, Jun. 2019, pp. 1954–1963, doi: 10.1109/CVPRW.2019.00247.

[55] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "TDAN: Temporally-deformable alignment network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 3357–3366, doi: 10.1109/CVPR42600.2020.00342.

[56] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multi-temporal ultra dense memory network for video super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2503–2516, Aug. 2020, doi: 10.1109/TCSVT.2019.2925844.

[57] T. Isobe, S. Li, X. Jia, S. Yuan, G. Slabaugh, C. Xu, Y.-L. Li, S. Wang, and Q. Tian, "Video super-resolution with temporal group attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8008–8017.

[58] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, "Video super-resolution with recurrent structure-detail network," in *Proc. IEEE Int. Conf. Eur. Conf. Comput. Vis. (ECCV)*, vol. 12357. Cham, Switzerland: Springer, 2020, pp. 645–660.

[59] M. A. A. Ali and M. A. Deriche, "Implementation and evaluate the no-reference image quality assessment based on spatial and spectral entropies on the different image quality databases," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. (ICoICT)*, May 2015, pp. 188–194.

[60] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," 2016, *arXiv:1611.06440*. [Online]. Available: http://arxiv.org/abs/1611.06440

[61] *Efficient Video Super-Resolution Via Hierarchical Temporal Residual Networks*. Accessed: Jul. 10, 2021. [Online]. Available: https://github.com/Holmes-Alan

ZHI-SONG LIU (Member, IEEE) received the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, under the supervision of Prof. Wan-Chi Siu and Dr. Yui-Lam Chan. He was a Postdoctoral Researcher with the Ecole Polytechnique, France, under the supervision of Prof. Marie-Paule Cani. He is currently a Postdoctoral Researcher with the Caritas Institute of Higher Education, Hong Kong. His research interests include machine learning and pattern recognition, image and video signal processing, and computing visions.

WAN-CHI SIU (Life Fellow, IEEE) received the M.Phil. degree from The Chinese University of Hong Kong, in 1977, and the Ph.D. degree from Imperial College London, in 1984. He was a Chair Professor, the Founding Director of the Signal Processing Research Centre, the Head of the Electronic and Information Engineering Department, and the Dean of Engineering Faculty, The Hong Kong Polytechnic University. He is currently an Emeritus Professor with The Hong Kong Polytechnic University. He is also a Research Professor with the Caritas Institute of Higher Education. He is an expert in DSP, transforms, fast algorithms, machine learning, deep learning, super-resolution imaging, 2D and 3D video coding, and object recognition and tracking. He has published 500 research articles (over 200 appeared in international journals), and edited three books. He has nine recent patents granted. He was an Independent Non-Executive Director (2000–2015) of a publicly-listed video surveillance company and a Convenor of the First Engineering/IT Panel of the RAE (1992–1993), Hong Kong. He is a fellow of IET. He is an outstanding scholar, with many awards, including the Best Teacher Award, the Best Faculty Researcher Award (twice), and the IEEE Third Millennium Medal (2000). He has been a Guest Editor/Subject Editor/AE of IEEE Transactions on Circuits and Systems—II: Express Briefs, IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, and *Electronics Letters*, and organized very successfully over 20 international conferences, including IEEE society-sponsored flagship conferences, such as a TPC Chair of ISCAS1997, a General Chair of ICASSP2003, and a General Chair of ICIP2010. He was the Past President (2017–2018) of Asia-Pacific Signal and Information Processing Association (APSIPA), and the Vice-President, a Chair of Conference Board, and a Core Member of Board of Governors (2012–2014) of the IEEE Signal Processing Society. He has been a member of the IEEE Educational Activities Board, the IEEE Fourier Award for Signal Processing Committee (2017–2020), the Hong Kong RGC Engineering Panel Member-JRS (2020–2022), and some other IEEE technical committees.

YUI-LAM CHAN (Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees from The Hong Kong Polytechnic University, Hong Kong, in 1993 and 1997, respectively.

He joined The Hong Kong Polytechnic University, in 1997, where he is currently an Associate Professor with the Department of Electronic and Information Engineering. He is actively involved in professional activities. He has authored over 110 research articles in various international journals and conferences. His research interests include multimedia technologies, signal processing, image and video compression, video streaming, video transcoding, video conferencing, digital TV/HDTV, 3DTV/3DV, multiview video coding, machine learning for video coding, and future video coding standards, including screen content coding, light-field video coding, and 360-degree omnidirectional video coding.

Dr. Chan serves as an Associate Editor for IEEE Transactions on Image Processing. He was the Secretary of the 2010 IEEE International Conference on Image Processing. He was also the Special Sessions Co-Chair and the Publicity Co-Chair of the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, and the Technical Program Co-Chair of the 2014 International Conference on Digital Signal Processing.