Learning a Deep Dual Attention Network for Video Super-Resolution

Feng Li, Huihui Bai, and Yao Zhao, *Senior Member, IEEE*

***Abstract*— Recently, deep learning based video super-resolution (SR) methods combine the convolutional neural networks (CNN) with motion compensation to estimate a high-resolution (HR) video from its low-resolution (LR) counterpart. However, most previous methods conduct downscaling motion estimation to handle large motions, which can lead to detrimental effects on the accuracy of motion estimation due to the reduction of spatial resolution. Besides, these methods usually treat different types of intermediate features equally, which lack flexibility to emphasize meaningful information for revealing the high-frequency details. In this paper, to solve above issues, we propose a deep dual attention network (DDAN), including a motion compensation network (MCNet) and a SR reconstruction network (ReconNet), to fully exploit the spatio-temporal informative features for accurate video SR. The MCNet progressively learns the optical flow representations to synthesize the motion information across adjacent frames in a pyramid fashion. To decrease the mis-registration errors caused by the optical flow based motion compensation, we extract the detail components of original LR neighboring frames as complementary information for accurate feature extraction. In the ReconNet, we implement dual attention mechanisms on a residual unit and form a residual attention unit to focus on the intermediate informative features for high-frequency details recovery. Extensive experimental results on numerous datasets demonstrate the proposed method can effectively achieve superior performance in terms of quantitative and qualitative assessments compared with state-of-the-art methods.**

***Index Terms*— Video super-resolution, motion compensation, detail components, attention mechanisms, high-frequency details.**

I. Introduction

VIDEO or multi-frame super-resolution (SR) is a classic problem in image processing, which aims at generating high-resolution (HR) frames from a given low-resolution (LR) video sequence. Video SR has been widely used in practical applications such as video surveillance, human face hallucination and video conversion.

Manuscript received June 23, 2019; revised November 25, 2019 and February 2, 2020; accepted February 3, 2020. Date of publication February 12, 2020; date of current version February 21, 2020. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2019JBZ102 and in part by the National Natural Science Foundation of China under Grant 61972023. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lisimachos P. Kondi. *(Corresponding author: Huihui Bai.)*

The authors are with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing Jiaotong University, Beijing 100044, China, and also with the Institute Information Science, Beijing Jiaotong University, Beijing 100044, China (e-mail: l1feng@bjtu.edu.cn; hhbai@bjtu.edu.cn; yzhao@bjtu.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.2972118

In the problem of video SR, a corrupted low-quality LR video is usually produced from the corresponding HR video via variant motion blurs, down-sampling operation and additive noise. We can observe that super-resolving the LR videos in real-world dynamics is an ill-posed problem since there are multitude of solutions to constrain irreversible degradations for any LR input. Various methods have been proposed to tackle such video SR problem can be divided into two categories: 1) single-frame SR, which mainly comes from single image SR [1]–[5]. This type of approach completely focuses on intra-frame spatial correlations and learns the mapping function from LR frames to HR frames individually. And 2) multi-frame SR [6]–[8] take the inter-frame temporal dependencies between consecutive LR frames into consideration to produce the HR frame.

Existing image SR algorithms can be roughly categorized into interpolation-based approach and example-based approach. The interpolation-based SR approach estimates the pixels in HR grid from an observed LR image via fixed weights, local covariance coefficients [12], and adaptive structure kernels [13], [14], which can simply achieve acceptable SR results but produce blurry edges and unpleasant artifacts. The example-based methods exploit the internal similarities of a same image [15]–[17] or learn the relationship between LR and HR image patches from external exemplar pairs [2], [18]–[21]. In recent years, with the significant improvement of deep learning in computer vision field, many methods [22]–[26] typically utilize convolutional neural networks (CNN) to directly learn the non-linear LR-to-HR mapping function for image SR and have achieved impressive performance on reconstruction accuracy and visual quality.

Multi-frame SR methods mainly focus on the inter-frame relations between consecutive LR frames. Most previous multi-frame SR methods [6], [27]–[29] model the temporal dependencies by conducting the sub-pixel motion registration based on sparse prior [27], [28] and total variation [6], [29]. Nevertheless, such iterative motion estimation can involve expensive computational cost and limit the capacity performing on large and complex motions. Recently, many methods combine the representation ability of deep learning with the inter-frame consistency to enhance the visual quality of HR frames. One option is to align adjacent frames as input to obtain the super-resolved center frame without explicit motion compensation [7], [8], which can reduce the computational cost caused by temporal alignment. Some other methods model the inter-frame correlations via bidirectional recurrent

architectures [30], [31], which learn the temporal dependencies without pre-/post-processing for multi-frame SR. However, these methods can produce HR images with visual artifacts on fast moving objects because of the non-explicit motion registration.

Most video SR algorithms [32]–[36] depend on the accurate motion estimation, which mainly consists of two steps, *i.e.* a motion estimation and compensation process followed by a SR reconstruction procedure. Some methods [32]–[34] use optical flow algorithms for motion registration and employ CNNs to model the non-linear mappings from compensated "cubes" to HR patches. Other methods [35]–[37] estimate the optical flow between consecutive frames with learned CNNs and produce HR frames through another deep networks, which can jointly conduct the motion compensation with SR task via an end-to-end trainable framework. In [38], Liu *et al.* introduce a temporal adaptive network to determine the optimal temporal scale and adaptively combine all the HR estimations based on motion information at pixel level.

However, all of these methods conduct the downscaling motion estimation via strided convolution to effectively handle large scale motions. Due to the reduction of spatial resolution, such approach can cause coarse optical flow representations and detrimental effects on motion estimation. Some methods [7], [35], [36] simply concatenate the compensated neighboring frames with center frame for SR reconstruction, which can suffer from the mis-registration errors caused by inaccurate motion estimation. Moreover, the LR inputs and features in deep CNNs contain different types information including low- and high-frequency components. The low-frequency components describe the main parts of images and the high-frequency components are responsible for the edge and texture details. Previous methods treat the information equally and lack flexibility to emphasize meaningful information for high-frequency details recovery.

In this work, we propose a novel deep dual attention network (DDAN), cascading a motion compensation network (MCNet) and a SR reconstruction network (ReconNet), to jointly exploit the spatio-temporal dependencies for video SR. The MCNet utilizes a pyramid motion compensation framework to learn multi-scale optical flow representations and further synthesize the motion information across adjacent frames in a coarse-to-fine manner. Besides the downscaling motion estimation as other methods, our MCNet employs an additional module without any downsampling operation to learn the full resolution optical flow representations for more accurate motion compensation. Then, instead of directly feeding the aligned frames and original center frame into ReconNet for SR reconstruction, we extract the detail components of original neighboring frames to mitigate the errors of motion estimation.

In the ReconNet, we present the residual attention group (RAG) that is composed of multiple residual attention blocks (RAB) to enhance the feature representation ability of our models for high-frequency details recovery. Specifically, we implement dual attention mechanisms, *i.e.* channel attention and spatial attention, on a residual block and form the RAB. The RAB can adaptively modulate intermediate features along channel and spatial dimensions to capture more important information within each feature map. At the end of ReconNet, an upscale module is employed to reconstruct the center HR residual frame from LR inputs. We further conduct the global residual learning between the HR residual image and bicubic amplified center frame to generate the HR frame.

In summary, the main contributions of this paper are summarized as follows:

1) We propose a novel deep dual attention network (DDAN) for video SR, which is composed of a motion compensation network (MCNet) and a SR reconstruction network (ReconNet), to fully exploit the spatio-temporal dependencies and learn more meaningful information for accurate video SR.
2) The MCNet investigates multi-level optical flow representations between adjacent frames in a pyramid fashion and infers the spatial transform between them to model the motion compensation.
3) We extract the detail components of original LR neighboring frames as complementary information for more accurate feature extraction, which can alleviate the mis-registration errors of motion estimation.
4) In the ReconNet, we combine dual attention mechanism along channel and spatial dimensions with residual learning to emphasize meaningful features for high-frequency details recovery. The MCNet and ReconNet can be jointly end-to-end trainable for motion compensation and video SR reconstruction.

The remainder of the paper is organized as follows. In Section II, we describe the related work. The details of our proposed video SR method are presented in Section III. We discuss the differences between our proposed residual attention mechanism and other attention based SR methods in Section IV. Ablation study and experimental comparisons with other state-of-the-art SR methods are provided in Section V. We conclude our work in Section VI.

II. Related Work

In this section, we first give a brief review of deep learning based image SR methods. Then we introduce the development of video SR. Finally, we discuss the attention mechanisms applied in deep neural networks.

A. Deep Learning Based Image Super-Resolution

Single image SR is a long-standing problem in computer vision, which refers to the transformation of an image from LR to HR. Since Dong *et al.* [5] utilize three-layer convolutional neural network for image SR (SRCNN), in recent years, deep learning methods have been widely used to tackle the ill-posed SR problem. Kim *et al.* [22] solve the image SR problem using a very deep convolutional network and residual learning (VDSR). In [24], Shi *et al.* introduce an efficient sub-pixel convolutional network (ESPCN) for image SR, which employs a sub-pixel convolutional layer to super-resolve the LR data into HR space at the end of the network. Lai *et al.* [25] present a deep laplacian pyramid SR Network (LapSRN) to progressively reconstruct the sub-band residues of HR images.

Tai *et al.* [26] propose a deep recursive residual network (DRRN) for image SR, which combines the recursive and residual learning strategy in global and local manners to mitigate the difficulty of training very deep network. In [39], Zhang *et al.* propose a residual dense network (RDN) to make full use of hierarchical features from the original LR image, which achieves the best state-of-the-art SR performance.

B. Video Super-Resolution

Video or multi-frame SR assumes that different observations of the same scene are available. Therefore, the shared explicit spatio-temporal redundancy can be used to constrain the SR problem and invert the downscaling process. In [6], Liu *et al.* propose a bayesian approach to simultaneously estimate the underlying motion, blur kernel, and noise level while reconstructing the HR frames. Ma *et al.* [27] tackle ubiquitous motion blurs by optimally searching the least blurred pixels for multi-frame SR. Motivated by the impressive performance of deep learning in image SR, most recent video SR methods adopt deep CNNs to directly learn the mappings from consecutive LR frames to HR frames. Kappeler *et al.* [7] apply an adaptive motion compensation scheme to handle fast moving objects and motion blurs in videos. Liao *et al.* [32] consider various motion information to generate an ensemble of SR drafts and then reconstruct HR frames by a draft-ensemble network. In [35], Caballero *et al.* present a real-time approach for video SR based on a sub-pixel layer and spatio-temporal network (VESPCN) to generate the HR frame from input LR consecutive frames. Based on the motion compensation transformer module in [35], Tao *et al.* [36] propose a sub-pixel motion compensation (SPMC) layer to align the reference frame to neighboring one onto HR grid. Then, an encoder-decoder network with a ConvLSTM [40] module is adopted to reconstruct the center HR image. Inspired by the motion compensation transformer module (MCT) in [35], [36], Wang *et al.* [41] propose a multi-memory convolutional neural network (MMCNN) for video SR, which utilizes serial densely connected residual blocks composed of multiple ConvLSTM layers to model the spatio-temporal correlations for video SR.

C. Attention Based Deep Models

Recently, attention mechanism has been proved to play an important role in capturing long-range dependencies in neural networks, which enables models to differentiate irrelevant information and focuses on more informative components of an input. The benefits of such a mechanism have been shown across a range of tasks, such as machine translation [42] and image classification [43]–[45]. Vaswani *et al.* [42] propose a deep model which entirely relies on an attention mechanism to draw global dependencies between input and output for machine translation. In [44], Hu *et al.* focus on the channel relationship and propose the squeeze and excitation network (SENet), which can adaptively recalibrate the channel-wise feature responses by explicitly modeling interdependencies between channels. Sanghyun *et al.* [45] introduce a convolutional block attention module (CBAM) which applies channel and spatial attention to emphasize meaningful features. There are also some methods [46]–[48] utilize channel attention or spatial attention to estimate HR images and show impressive SR performance.

III. Proposed Method

In this section, we present the design methodology for our proposed DDAN. We first introduce the whole architecture of our network, and then the details of each individual module are provided.

A. Overview

The degradation procedure of a HR video sequence to the corrupted low-quality sequence at time t can be represented as $I_t^L = (B_t \otimes I_t^H) \downarrow_s + \epsilon_t$. Here, I_t^H denotes the clean HR frame at time t and I_t^L is the corresponding center LR frame via multiple quality degradations. B_t represents the complex motion variations such as motion blur and defocus blur. $\downarrow_s$ is the downsampling operation with scale factor s and ϵ_t is the additive noise. The pixel-level motion registration between the i^{th} neighboring frame I_i^H and center frame I_t^L can be formulated as $I_t^L = BC_{i,t}(I_i^H) \downarrow_s + \epsilon_{i,t}$. Here, $C_{i,t}(\cdot)$ is the warping operation for aligning I_i^H to I_t^H. B denotes the blur matrix. $\epsilon_{i,t}$ contains the additive noise and mis-registration errors.

Given a corrupted video sequences $\{I_i^L\}_{t-N}^{t+N}$, the goal of our proposed DDAN is to generate the center HR frame $\hat{I}_t^H \in \mathbb{R}^{sH \times sW \times C}$ from the center LR frame $I_t^L \in \mathbb{R}^{H \times W \times C}$ and $2N$ neighboring frames $[I_{t-N}^L, \ldots, I_{t-1}^L, I_{t+1}^L, \ldots, I_{t+N}^L]$ with scale factor s. As shown in Fig. 1, the proposed network is composed of two parts: a motion compensation network (MCNet) and a SR reconstruction network (ReconNet). The MCNet network takes the center frame I_t^L and neighboring frame I_i^L as input to produce the motion compensated neighboring frame $\hat{I}_i^L$

$$\hat{I}_i^L = F_{MC}(I_t^L, I_i^L) \tag{1}$$

where $F_{MC}(\cdot)$ denotes the mapping function from I_t^L to I_i^L of MCNet. Then, to further alleviate the mis-registration influence of motion estimation, we extract the detail components d_i by calculating the residues between the aligned frame $\hat{I}_i^L$ and its corresponding input I_i^L

$$d_i = I_i^L - \hat{I}_i^L \tag{2}$$

Feeding $2N$ neighboring frames into the MCNet, we can obtain $2N$ aligned LR frames $[\hat{I}_{t-N}^L, \ldots, \hat{I}_{t-1}^L, \hat{I}_{t+1}^L, \ldots, \hat{I}_{t+N}^L]$ and the detail components $[d_{t-N}, \ldots, d_{t-1}, d_{t+1}, \ldots, d_{t+N}]$. In the ReconNet, we concatenate the aligned frames, resulted detail components and LR center frame in channel dimension and take them as input of ReconNet for feature extraction and SR reconstruction

$$\hat{I}_t^H = F_{SR}(I_t^L, \hat{I}_i^L, d_i) \tag{3}$$

where $F_{SR}(\cdot)$ denotes the mapping function of ReconNet to reconstruct the HR center frame.

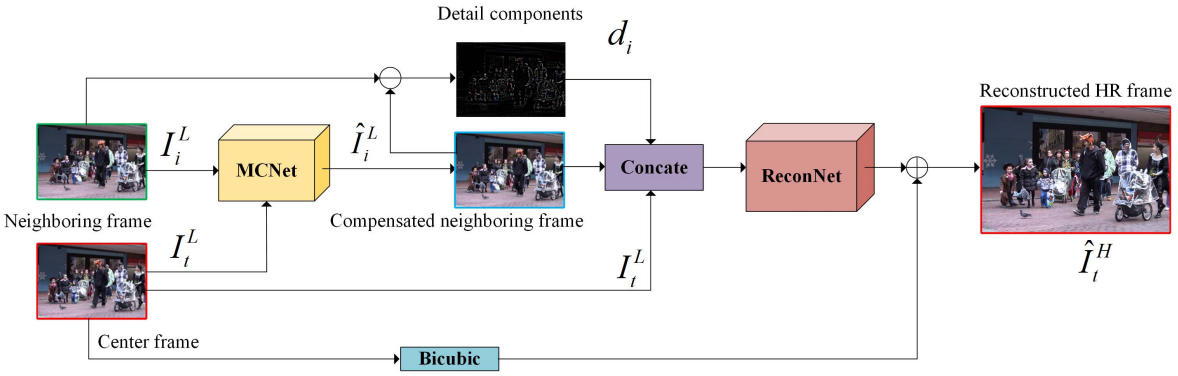


Fig. 1. The network architecture of our proposed deep dual attention network (DDAN) for spatio-temporal video SR, which contains a motion compensation network (MCNet) to synthesize the motion information across the neighboring frames at different scales and a SR reconstruction network (ReconNet) to generate HR frames.

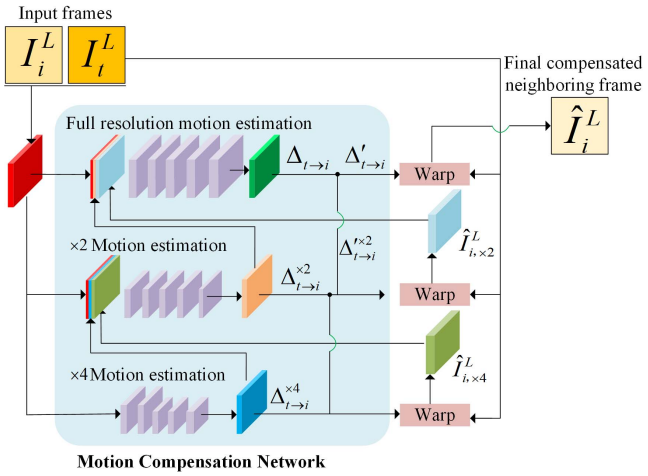


Fig. 2. The network architecture of our proposed motion compensation network (MCNet), which adopts a pyramid architecture to learn multi-level optical flow representations and synthesize the motion information across neighboring frames at different scales.

B. Motion Compensation Network

Previous methods [35]–[37], [41] learn the downscaling flow representations to model the motion compensation. Although such approaches can effectively handle large motions, with the first down-scaling operations, the reduction of spatial resolution can cause detrimental effects on the accuracy of motion estimation. In order to obtain more accurate aligned frames, in the proposed MCNet, besides learning the downscaling flow representations, we develop an additional motion estimation module to learn the full resolution optical flow representations. As sketched in Fig. 2, we adopt a pyramid multi-level structure to conduct the motion compensation between adjacent frames. To simply depict the details of our MCNet, we introduce the motion compensation strategy between two frames. Given original LR input frames I_i^L and I_t^L, a $\times 4$ coarse optical flow $\Delta^{\times 4}_{t\to i}$ is obtained by early concatenating the two frames and then downscaling with two $\times 2$ strided convolutional layers. The estimated optical flow is upscaled to original resolution by a sub-pixel convolutional layer. A schematic design of the motion estimation modules

TABLE I

THE ARCHITECTURE OF THE MULTI-LEVEL MOTION ESTIMATION MODULES IN THE PROPOSED MCNET. CONVOLUTIONAL LAYERS ARE DESCRIBED BY KERNEL KIZE (k), STRIDE (s), AND NUMBER OF CHANNELS (n)

Layers	$\times 4$ flow	$\times 2$ flow	Full resolution flow
1	k5s2n24 / ReLU	k5s2n24 / ReLU	k3s1n32 / ReLU
2	k3s1n24 / ReLU	k3s1n24 / ReLU	k3s1n32 / ReLU
3	k5s2n24 / ReLU	k3s1n24 / ReLU	k3s1n32 / ReLU
4	k3s1n24 / ReLU	k3s1n24 / ReLU	k3s1n32 / ReLU
5	k3s1n32 / tanh	k3s1n8 / tanh	k3s1n2 / tanh
6	Upscale $\times 4$	Upscale $\times 2$	—

are detailed described in Table I. The resulted coarse flow $\Delta^{\times 4}_{t\to i}$ is applied to warp the target frame and produce $\hat{I}^L_{i,\times 4}$. Motivated by [35] and [49], we adopt the bilinear interpolation for more efficient warping process

$$\hat{I}^L_{i,\times 4} = \mathcal{I}(I_t^L, \Delta^{\times 4}_{t\to i}) \tag{4}$$

where $\mathcal{I}(\cdot)$ represents the warping operation. The warped frame $\hat{I}^L_{i,\times 4}$ is then processed with the coarse flow $\Delta^{\times 4}_{t\to i}$ and the LR center frame I_t^L through a $\times 2$ motion estimation module. As shown in Table I, this uses only one $\times 2$ strided convolutional layers and a $\times 2$ upscale layer to obtain a $\times 2$ optical flow $\Delta^{\times 2}_{t\to i}$. We then obtain the finer aligned frame $\hat{I}^L_{i,\times 2}$ by warping the center frame with the combined optical flow $\Delta'^{\times 2}_{t\to i}$

$$\begin{aligned} \Delta'^{\times 2}_{t\to i} &= \Delta^{\times 2}_{t\to i} + \Delta^{\times 4}_{t\to i} \\ \hat{I}^L_{i,\times 2} &= \mathcal{I}(I_t^L, \Delta'^{\times 2}_{t\to i}) \end{aligned} \tag{5}$$

We further utilize an additional motion estimation (the top branch in Fig. 2) which contains several convolutional layers without any down-scaling process to learn the full resolution optical flow representations. As shown in Fig. 2, the output optical flow $\Delta'^{\times 2}_{t\to i}$ and corresponding compensated frame $\hat{I}^L_{i,\times 2}$ are fused with the original LR center frame I_t^L as the input of our full resolution motion estimation module. Then, the full resolution flow $\Delta_{t\to i}$ can be generated. Therefore, we can obtain the final compensated neighboring frame $\hat{I}_i^L$

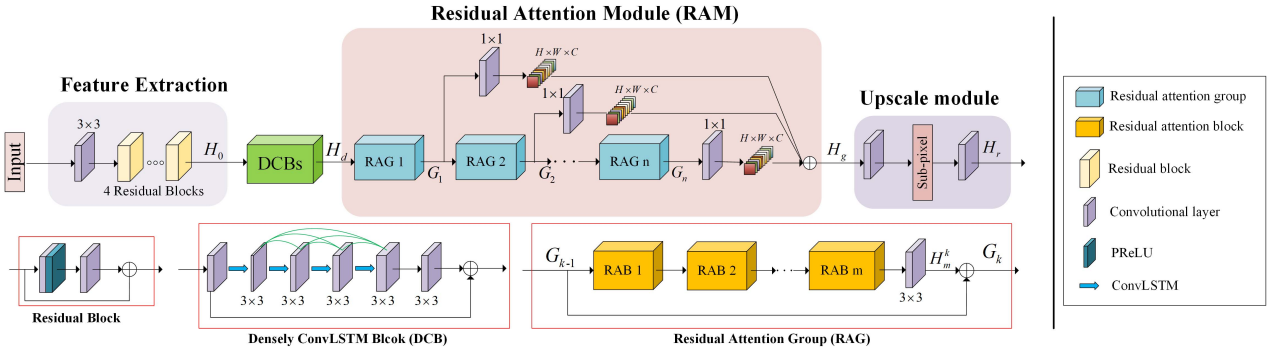


Fig. 3. The network architecture of our proposed SR reconstruction network (ReconNet), which mainly contains a feature extraction module, multiple stacked densely ConvLSTM blocks as build module, a residual attention module (RAM) composed of multiple residual attention groups (RAG), and a upsample module to upscale the LR inputs to desired spatial resolution.

with the total flow $\Delta'_{t\to i}$

$$\begin{aligned}\Delta'_{t\to i} &= \Delta'^{\times 2}_{t\to i} + \Delta_{t\to i}\\ \hat{I}_i^L &= \mathcal{I}(I_t^L, \Delta'_{t\to i})\end{aligned} \tag{6}$$

C. Detail Components Extraction

Previous sophisticated optical flow based methods [35], [36], [41] simply concatenate the compensated neighboring frames and center frame for feature extraction and reconstruction. However, any errors in the optical flow estimation or wrapping operation can adversely affect the subsequent SR reconstruction and introduce artifacts. To solve this issue, as shown in Fig. 1, we extract the detail components of neighboring frames by conducting the subtraction operation between the aligned frames and their original LR inputs. After that, the extracted detail components are concatenated with the warped frames and center LR input in channel dimension. For simply depiction, we formulate such fusion step as

$$\mathbf{I^f} = [I_t^L, \hat{I}_i^L, d_i] \tag{7}$$

where $\mathbf{I^f}$ denotes the concatenated input of the three components.

D. SR Reconstruction Network

The detailed structure of the SR reconstruction network (ReconNet) is shown in Fig. 3. The proposed ReconNet contains four parts: a feature extraction module, multiple stacked densely ConvLSTM blocks (DCB) as build module, a residual attention module (RAM) and an upscale module.

1) Feature Extraction: As illustrated in Fig. 3, the feature extraction module contains a 3×3 convolutional layer and serial residual blocks composed of two convolutional layers with learnable kernel of size 3×3 to extract deep features from the fused input $\mathbf{I^f}$ fed into ReconNet

$$H_0 = h_0(\mathbf{I^f}) \tag{8}$$

where $h_0(\cdot)$ denotes the mapping function of the feature extraction module. H_0 represents the extracted features and is used as input to later state.

2) Densely ConvLSTM Blocks: Recent video SR methods [36], [41] employ ConvLSTM [40] to exploit the inter-frame correlations of input video sequences and have generated promising SR results. Specifically, supposing there are inputs $\mathcal{X}_1, \ldots, \mathcal{X}_t$, cell outputs $\mathcal{C}_1, \ldots, \mathcal{C}_t$, hidden states $\mathcal{S}_1, \ldots, \mathcal{S}_t$, and the input gate i_t, output gate o_t, forget gate f_t of the ConvLSTM, the key equations of ConvLSTM are shown as below

$$\begin{aligned}i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{si} * \mathcal{S}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1})\\ f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{sf} * \mathcal{S}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1})\\ \mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ tanh(W_{xc} * \mathcal{X}_t + W_{sc} * \mathcal{S}_{t-1})\\ o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{so} * \mathcal{S}_{t-1} + W_{co} \circ \mathcal{C}_t)\\ \mathcal{S}_t &= o_t \circ tanh(\mathcal{C}_t)\end{aligned} \tag{9}$$

where $\sigma(\cdot)$ and $tanh(\cdot)$ denote the sigmoid and hyperbolic tangent function. "* " denotes the convolution operator and "$\circ$" denotes the Hadamard product. We can see that a ConvLSTM can capture motions when we view the states as the hidden representations of moving objects. We have tested two ConvLSTM methods DRVSR [36] and MMCNN [41] for video SR. We find that multiple "Conv-ConvLSTM" with densely connections can effectively model the temporal dependencies and shows better performance on validation datasets during the training process. Therefore, in our method, as illustrated in Fig. 3, we employ multiple densely ConvLSTM Blocks and insert them in the middle stage of ReconNet to make full use of spatio-temporal information. This process can be represented as

$$H_d = h_D(h_0) \tag{10}$$

where $h_D(\cdot)$ denotes the mapping function of the whole DCBs module and H_d denotes the learned features.

3) Residual Attention Module: The LR inputs and features in deep CNN contain different types information such as low- and high-frequency information. The low-frequency components describe the main parts of images and the high-frequency components are responsible for the edge and texture details. Thus, to make our proposed network emphasize more meaningful information, as shown in Fig. 3, in our proposed residual attention module (RAM), we utilize multiple residual attention groups (RAG) to exploit the interdependencies among

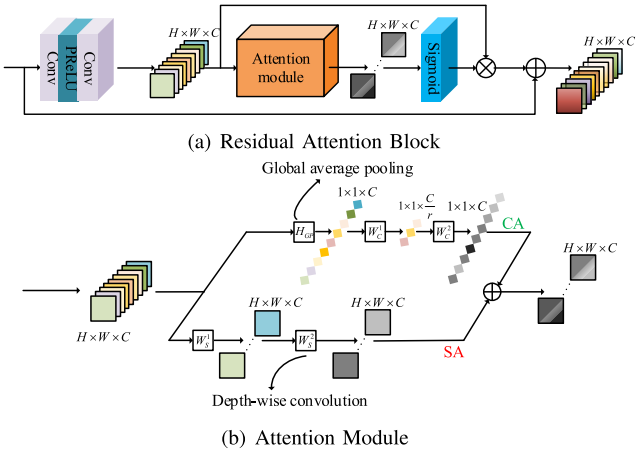


Fig. 4. The structure of the residual attention block in the proposed RAG. Top: the full components in the RAB. Bottom: the detailed design of the attention module (AM) in the RAB, which contains one channel attention (CA) unit (top branch) and one spatial attention (SA) unit (bottom branch).

inter-channel and spatial dimensions. The RAG consists of serial residual attention blocks (RAB) followed by a 3×3 convolutional layer. The RAB combines the classical residual unit [50] with spatial and channel attention mechanisms. Supposing there are n RAGs in ReconNet and each RAG contains m RABs. Therefore, the output G_n of the n^{th} RAG can be represented as

$$\begin{aligned} G_n &= G_{n-1} + h_m^n(\mathcal{R}_m^n(...(\mathcal{R}_2^n(\mathcal{R}_1^n(G_{n-1})))...)) \\ &= G_{n-1} + H_m^n \end{aligned} \tag{11}$$

where G_{n-1} is the output of the $(n-1)^{th}$ RAG and the input of the n^{th} RAG. $[\mathcal{R}_2^n, \mathcal{R}_1^n, \ldots, \mathcal{R}_{m-1}^n, \mathcal{R}_m^n]$ denote the mapping functions of m RABs in the n^{th} RAG. $h_m^n(\cdot)$ denotes the convolution operation of the final convolutional layer and H_m^n is the output via the convolution operation.

4) Residual Attention Block: Now, we elaborate the details of the residual attention block (RAB) in our proposed RAG. As shown in Fig. 4(a), each RAB contains two 3×3 convolutional layers and one attention module (AM). For the j^{th} RAB, the output U_j of the first two convolutional layers can be represented as

$$U_j = f_j^2(f_j^1(R_{j-1})) \tag{12}$$

where R_{j-1} is served as the output of the $(j-1)^{th}$ RAB and the input of the j^{th} RAB. $f_j^1(\cdot)$ and $f_j^2(\cdot)$ are the mapping functions of the two convolutional layers, respectively. We denote the $U_j = [u_j^1, u_j^2, \ldots, u_j^{C-1}, u_j^C]$ consisting of C feature maps with the size $H \times W \times C$ as the input of our AM. The structure of the AM in RABs is illustrated in Fig. 4(b). We embed the spatial attention (SA) unit and channel attention (CA) unit to exploit the interdependencies of features between the channels and spatial locations. As shown in Fig. 4(b) (the top branch), for the CA unit, we first conduct global average pooling operation on U_j to obtain the channel-wise statistic $z \in \mathbb{R}^{1\times1\times C}$ through spatial dimensions $H \times W$

$$z_c = f_{GP}(U_j) = \frac{1}{H \times W} \sum_{p=1}^{H} \sum_{q=1}^{W} u_j^c(p, q) \tag{13}$$

where $u_j^c(p, q)$ is the value at position (p, q) of the c^{th} channel u_j^c. $f_{GP}(\cdot)$ denotes the global pooling operation. To fully capture the interdependencies across channels from the aggregated information by global average pooling, we employ two 1×1 convolutional layers with the reduction ratio r to extract the summary statistic $\mathbf{z}$

$$M_j^{CA} = W_C^2 * \tau(W_C^1 * \mathbf{z}) \tag{14}$$

where M_j^{CA} denotes the resulted channel attention map of the j^{th} RAB. $W_C^1 \in \mathbb{R}^{\frac{C}{r}\times C\times1\times1}$ and $W_C^2 \in \mathbb{R}^{C\times\frac{C}{r}\times1\times1}$ are the weight sets of the two 1×1 convolutional layers in CA unit, respectively. $\tau(\cdot)$ is the PReLU [51] activation function and "$*$" denotes the convolution operation.

Different from CA, the SA focuses on more important regions and model the contextual information over local representations. Given the same input $U_j = [u_j^1, u_j^2, \ldots, u_j^{C-1}, u_j^C]$ for the SA unit, as shown in Fig. 4(b) (the bottom branch), we first adopt a 1×1 convolutional layer to integrate the features of previous state. Then, one depth-wise convolutional layer is employed to obtain different spatial attention maps for each channel

$$M_j^{SA} = W_S^2 * \tau(W_S^1 * \mathbf{z}) \tag{15}$$

where $M_j^{SA} \in \mathbb{R}^{H\times W\times C}$ denotes the generated spatial attention maps. $W_S^1 \in \mathbb{R}^{H\times W\times C}$ and $W_S^2 \in \mathbb{R}^{H\times W\times C}$ represent the weight sets of the first convolutional layer and the following depth-wise convolutional layer in SA unit, respectively.

To take advantage of the both attention mechanisms simultaneously, we conduct the element-wise addition operation on the attention maps produced via CA unit and SA unit. After that, we utilize a sigmoid function to normalize such attention maps range to $[0, 1]$ for generating a full attention mask $\gamma \in \mathbb{R}^{H\times W\times C}$

$$\gamma = \sigma(M_j^{CA} + M_j^{SA}) \tag{16}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Thus, the output R_j of the j^{th} RAB can be formulated as

$$\begin{aligned} R_j &= R_j(R_{j-1}) \\ R_j &= R_{j-1} + \gamma \otimes U_j \end{aligned} \tag{17}$$

With the integration of CA and SA in residual blocks, the features are adaptively modulated in a global and local way to enhance the representational ability of our proposed network for high-frequency details recovery. Furthermore, to explore the features at different states, we take all output feature maps of the RAGs as input fed into a 1×1 convolutional layer respectively and generate a fusion representation. This process can be expressed as

$$H_g = \sum_{k=1}^{n} \boldsymbol{W_k} * G_k \tag{18}$$

where $\boldsymbol{W_k}$ denotes the weight set of the 1×1 convolutional layer for the k^{th} RAG and H_g is the fused representation.

5) Upscale Module: After extracting deep features in LR space, as illustrated in Fig. 3, a 3×3 convolutional layer with s^2C channels followed by a sub-pixel convolutional layer [35] is adopted to convert multiple LR subimages of size $H \times W \times s^2C$ to HR subimages of size $sW \times sH \times C$. Then a single channel convolutional layer with kernel size of 3×3 to reconstruct the HR residual image

$$H_r = h_r(\mathcal{PS}(h_u(H_g))) \tag{19}$$

where $h_u(\cdot)$ denotes the convolution operation to extract s^2C feature maps for the upscaling process. $\mathcal{PS}$ is an periodic shuffling operator that rearranges the elements of a $H \times W \times s^2C$ tensor to a tensor of shape $sW \times sH \times C$. $h_r(\cdot)$ denotes the mapping function of the reconstruction layer. H_r is the HR residual frame produced by ReconNet. Finally, as sketched in Fig. 1, global residual learning is conducted on the residue between the estimated HR residual frame H_r and bicubic amplified center frame to produce the final SR result. The output $\hat{I}_t^H$ of our proposed DDAN can be expressed as

$$\hat{I}_t^H = H_r + \mathcal{B}(I_t^L) \tag{20}$$

where $\mathcal{B}(\cdot)$ is the bicubic upsampling operation.

E. Training Strategy

Our proposed DDAN combines the MCNet and ReconNet to provide an accurate, fast, jointly end-to-end trainable motion compensated video SR method. Since we do not have the ground truth of optical flow, to train the MCNet for motion compensation, we adopt the unsupervised warping loss as [36] to optimize its parameter set Θ_1 and minimize the MAE between the compensated frame $\hat{I}_i^L$ and original neighboring frame I_i^L according the flow $\Delta'_{t\to i}$

$$\mathcal{L}_{mc}(\Theta_1) = \sum_{i=-N}^{N} ||I_i^L - \hat{I}_i^L||_1 + \alpha||\nabla_{t\to i}||_1 \tag{21}$$

where $\mathcal{L}_{mc}(\cdot)$ denotes the loss function of MCNet. $\nabla_{t\to i}$ denotes the total variation on each component of the learned optical flow $\Delta'_{t\to i}$ in MCNet. α is the regularization weight. We set $\alpha = 0.01$ in all experiments. The Charbonnier loss [25] is applied on the output of ReconNet and backpropagated through both ReconNet and MCNet

$$\mathcal{L}_{sr}(\Theta_2) = \sum_{i=-N}^{N} \sqrt{(I_t^H - \hat{I}_t^H)^2 + \boldsymbol{\varepsilon}^2} \tag{22}$$

where $\mathcal{L}_{sr}(\cdot)$ and Θ_2 represent the loss function and learned parameters of ReconNet, respectively. $\hat{I}_t^H$ is the final reconstructed HR frame of DDAN. I_t^H is the corresponding HR ground truth of I_t^L. We empirically set $\boldsymbol{\varepsilon}$ to 10^{-3}. Consequently, the overall loss function $\mathcal{L}(\cdot)$ to train DDAN is

$$\mathcal{L} = \mathcal{L}_{sr} + \beta\mathcal{L}_{mc} \tag{23}$$

where β is the non-negative trade-off weight. We set $\beta = 0.01$ in all experiments.

IV. Discussion

In this section, we discuss the differences between our proposed residual attention mechanism and other attention based SR methods which contains the convolutional block attention module (CBAM) [45], residual attention module (RAMSR, for a better distinction with our proposed RAM) in [47], and the channel-wise and spatial attention residual (CSAR) block [48].

A. Difference to CBAM

Given an intermediate feature map, CBAM [45] sequentially infers the channel attention and spatial attention for adaptive feature refinement. Compared to our proposed RAB, CBAM adopts both average and max pooling operations to obtain two type of channel statistics for finer attention while our proposed RAB only adopts average pooling to calculate the channel-wise statistics. Besides, the CBAM further introduces a spatial attention module to aggregate the spatial information, which also employs the average and max pooling followed by a large kernel size (7×7) of convolutional layer to obtain a single spatial attention map. Our RAB employs a 1×1 convolutional layer to integrate the features of previous state and one 3×3 depth-wise convolutional layer to obtain different spatial attention maps for each channel, which is more effective than CBAM for video SR. The comparisons will be shown in Table V and Fig. 9.

B. Difference to RAMSR and CSAR

We now elaborate the differences between our proposed RAM and RAMSR [47]. Different from our proposed RAM utilizing global average pooling to capture the channel-wise statistics, RAMSR computes the variances from each channel to extract channel-specific statistics. Compared to CSAR [48], we all utilize the average pooling operation on individual feature channel along spatial dimensions. However, in the spatial attention unit, CASR calculates a single spatial attention map to emphasize different local regions whereas our SA unit in RAB obtains different spatial attention maps for each channel, which can adaptively modulate the contextual information in a more effective way. The comparisons with RAMSR and CSAR will be shown in Table V and Fig. 9.

V. Experiments

In this section, we first introduce the datasets to train and test our models. Then, the implementation details of our proposed framework is provided. Next, we analyze different modules in our proposed network. Finally, our results are compared with several state-of-the-art methods in terms of quantitative evaluations, visual quality, and inference time.

A. Datasets

In this work, since there is no publicly available video dataset that is large enough to train our deep networks, we use the datasets provided in [41] as our training datasets, which contains 542 video sequences collected from high-quality videos with the contents including urban, wildlife, and landscape *et al.*. Each video sequence contains 32 consecutive

frames, where most frames at the resolution of 1280×720. We randomly select 522 video sequences as training data and the rest 20 for validation (termed as *Val20*). For testing, to demonstrate the effectiveness and generalization of our DDAN, we first conduct experiments on public single image testing datasets including *Set5* [52], *Set14* [53], *BSDS100* [54], *Urban100* [17] and *Manga109* [55] and compare our DDAN with recent image SR methods like A+ [3], SRCNN [5], VDSR [22], DRCN [23] and LapSRN [25]. We further compare our method with recent state-of-the-art video SR methods on three public video datasets: *Myanmar* [7], *Vid4* [35], and *YUV21* [33]. Original *Myanmar* video contains 59 scenes with 4K resolution (3840×2160), where 6 scenes composed of 4 frames are utilized for testing. The original frames are downscaled to 960×540 pixels as HR frames using bicubic interpolation. The *Vid4* dataset contains four videos: calendar, city, foliage, and walk, which are of 720×576, 704×576, 720×480, and 720×480 resolutions respectively. The *YUV21* dataset includes 21 CIF format clips with different types of motions captured in different scenes and all the videos are of 352×288 resolution. PSNR and SSIM are used as evaluation metrics to compare with different image and video SR networks quantitatively. In order to avoid the border effects, PSNR and SSIM are calculated by eliminating 8 pixels on each border as in [41].

B. Implementation Details

The detailed architecture of MCNet is illustrated in Table I and Fig. 2. With respect to the ReconNet, there are 4 residual blocks in the feature extraction module. We adopt 10 DCBs and each DCB contains four convolutional layers with the kernel size of 3×3 from 16 to 64 filters, including those inside ConvLSTM. In each RAG, all convolutional layers have 64 filters and the kernel size of them are 3×3 except the 1×1 convolutional layers in the AM. The reduction ratio r in CA unit is set to 16. The kernel size of depth-wise convolutional layer in SA unit is set to 3×3. In the upscale module, we adopt one 3×3 convolutional layer with $64s^2$ filters to integrate previous LR features for scale factor s ($s = 2, 3, 4$). The reconstruction layer at the end of our DDAN contains one filter with kernel size of 3×3 and stride 1.

We convert all the video frames into YCbCr color space, and send only the luminance channel to our network. All original LR input frames are downsampled with specific scale factors using the bicubic interpolation. The input LR frames fed into the proposed network are with the patches of 32×32 pixels from N_F consecutive frames with non-overlapping regions and the mini-batch size is set to 10. We initialize the network with Xavier method [56] and train our models using Adam [57] optimizer. The initial learning rate is set to $5e-4$ for all layers and decreases to $1e-5$ after 10^6 iterations using the Polynomial decay. We first train the MCNet for 10^5 iterations using $\mathcal{L}_{mc}$, then we train the ReconNet for another 10^5 iterations using $\mathcal{L}_{sr}$. At last, these two networks are trained together with $\mathcal{L}$ for 10^6 iterations. We use Tensorflow to implement our models on a Titan Xp GPU.

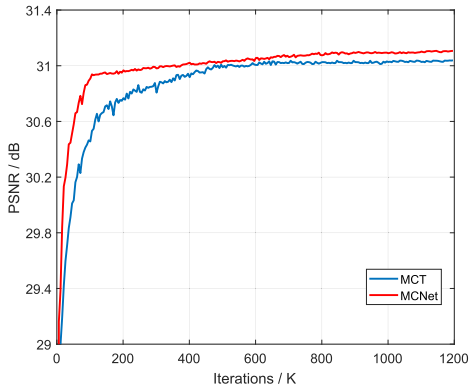


Fig. 5. The convergence process of the two motion compensation algorithms, MCT and MCNet, in the proposed DDAN. The curves for each combination are based on the PSNR for 4× SR on the *Val20* dataset.

(a) Ground truth (b) MCT [40] (c) MCNet (ours)

Fig. 6. The visual comparisons of video SR results by the networks with the combinations of MCT [41] and our MCNet. The assessments are made for 4× upscaling on the 12^{th} frame from "*foliage*" (top) and the 10^{th} frame from "*walk*" (bottom) video clips in *Vid4* dataset.

C. Investigation of the Motion Compensation Network

We compare the proposed MCNet with the motion compensation transformer module (MCT) in [35], [41]. MCT learns the $\times 4$ and $\times 2$ optical flow for motion estimation. Our proposed MCNet can be regarded as an extensive vision of MCT, which employs an additional full resolution motion estimation module to make more accurate motion compensation. We fixed the number of RAGs in ReconNet as 4 and each RAG contains 4 RABs. Then, we combine the two motion compensation strategies with our ReconNet respectively to investigate the effectiveness of different motion compensation algorithms for video SR. The convergence process of the two combinations is visualized in Fig. 5. We can observe that the proposed MCNet can stabilize the training process and achieve higher PSNR performance (about 0.15dB) than MCT with the same training time cost. Besides, to demonstrate that the superiority of our proposed MCNet for more accurate HR frame reconstruction, we illustrate the HR frames generated by the two combinations. As shown in Fig. 6, we can see that the proposed MCNet produces clearer image details while the model using MCT generate SR results with more blurs. Thus, we adopt MCNet as our motion compensation strategy and combine it with the ReconNet (4 RAGs and 4 RABs), termed as DDAN-M4N4.

(a) Ground truth (b) RB+DC (c) SAB+DC (d) CAB+DC (e) RAB+DC (f) RAB w/o DC

Fig. 7. The visual comparisons of video SR results by the networks with various combinations of the components in Table II. The assessments are made for 4× upscaling on the 6^{th} frame from "*foreman*" (top) and "*news*" (bottom) video clips in *YUV21*, where "DC" denotes detail components.

TABLE II

STUDY OF CA AND SA FOR VIDEO SR WITH SCALE FACTOR 4 ON *Val20*. THE **TEXT** INDICATE THE BEST PERFORMANCE

	Components	Different combinations of CA and SA				
		RB	CAB	SAB	RAB	Baseline
In residual blocks	CA	×	✓	×	—	—
	SA	×	×	✓	—	—
	CA and SA	×	—	—	✓	—
PSNR		30.96	31.03	31.02	**31.10**	30.52

TABLE III

ABLATION STUDY OF DETAIL COMPONENTS FOR DIFFERENT VIDEO SR MODELS. WE OBSERVE THE BEST PERFORMANCE ON *Val20* FOR 4× SR

Models	VESPCN / VESPCN-C	MMCNN / MMCNN-C	DDAN / DDAN-C
PSNR	29.32 / **29.50**	30.76 / **30.89**	30.91 / **31.10**
SSIM	0.7061 / **0.7128**	0.7440 / **0.7487**	0.7489 / **0.7519**

D. Study of Channel Attention, Spatial Attention

To validate the effectiveness of the proposed RAB for video SR, besides the RAB, we construct another three residual blocks with different attention mechanisms for comparison.

(i) CA based residual block (CAB): we remove the SA unit from RAB. Therefore, the CAB contains the two 3×3 convolutional layers and one CA unit.

(ii) SA based residual block (SAB): we remove the CA unit from RAB and build the SAB.

(iii) Basic residual block (RB): we remove the two attention mechanisms (*i.e.* CA and SA) from RAB and only retain the two 3×3 convolutional layers.

We adopt the network without any RAB or RB as a baseline model. Table II shows the ablation study on the effects of the channel attention (CA) and spatial attention (SA) for 4× SR video SR on *Val20*. The first 4 networks adopt the same structure as DDAN-M4N4. Obviously, we can see that the baseline model achieves the worst performance. This is because that the baseline model only adopts the same feature extraction module, DCBs and upscaling module for video SR, which has much fewer layers and parameters than another four models. From the comparisons of the first 4 networks, we can see that when both CA unit and SA unit are removed in the RAB, the PSNR values are relatively low. Besides, by integrating the CA unit or the SA unit into the residual blocks, the SR performances can be moderately improved. Finally, when our proposed RAB with the combinational two attentions CA and SA are utilized, the performance can be further boosted.

To demonstrate that the proposed RAB can help to produce more accurate high-frequency details, we show the visual comparisons of 4× SR results produced by the first 4 SR models in Table II on the *foreman* and *news* video sequences from *YUV21*. In Fig. 7, it is seen that the network with proposed model with RAB (Fig. 7(e)) can produce clearer image contents than the non-attention SR model (Fig. 7(b)).

E. Effectiveness of Detail Components

We now analyze the effects of the detail components extracted from neighboring frames for HR center frame recovery. The visual comparisons for 4× SR are visualized in Fig. 7. As illustrated in Fig. 7(f), compared with Fig. 7(e), we can observe that the model which fuses neighboring detail components can produce the SR results with more accurate texture details while the SR model without detail components extraction caused more blurs and artifacts. Then we conduct the experiments on two existing video SR methods VESPCN [35] and MMCNN [41]. Specifically, after the motion compensation via MCT, we extract the detail components as our DDAN. We retrain the pubic models VESPCN and MMCNN for fair comparison. The models trained with detail components are denoted as VESPCN-C and MMCNN-C. We adopt the DDAN-M4N4 structure (simply denoted as DDAN and DDAN-C) to compare with other models. Table III shows the quantitative comparisons on *Val20* for 4× SR. It is seen that the models trained with details components can obtain better PSNR and SSIM scores than their original models. Both the quantitative and qualitative results demonstrate that the detail components can dramatically improve the video SR performance.

F. Effectiveness of Residual Attention Module

In this subsection, we first investigate the basic network parameters: the number of RAG (denote as n for short),

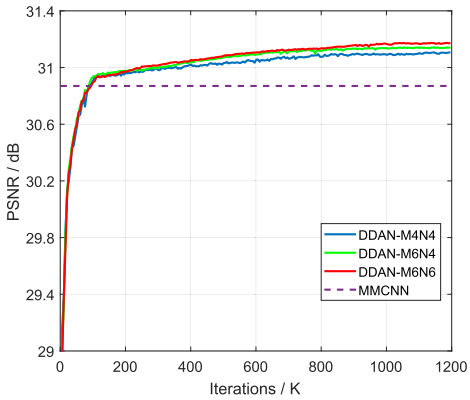


Fig. 8. The training process of our proposed DDAN with different number of n and m in the ReconNet. The curves for each model are based on the PSNR on *Val20* with scale factor 4 in 1200K iterations.

TABLE IV

ABLATION STUDY OF DCBS AND RAM. WE OBSERVE THE BEST PERFORMANCE ON *Val20* WITH SCALE FACTOR 4. THE **TEXT** INDICATE THE BEST PERFORMANCE

Models	Depth	Parameters (M)	PSNR (dB)
MMCNN	10 DCBs (100 layers)	10.582	30.87
DDAN-S	7 DCBs (141 layers)	**7.051**	31.11
DDAN	10 DCBs (**158 layers**)	10.290	**31.17**

TABLE V

ABLATION STUDY OF DIFFERENT ATTENTION MECHANISMS. WE OBSERVE THE BEST PERFORMANCE ON *Myanmar* DATASETS FOR 4× SR

Models	CBAM	RAMSR	CSAR	RAM (ours)
PSNR	34.29	34.38	31.35	**34.40**
SSIM	0.9105	**0.9134**	0.9132	**0.9134**

(a) Ground truth (b) CBAM (c) RAMSR (d) CSAR (e) RAM (ours)

Fig. 9. The visual comparisons of Different attention mechanisms implemented on our DDAN-S. The assessments are made for 4× upscaling on the 15^{th} frame from "*city*" video clips in *Vid4* datasets.

TABLE VI

INVESTIGATION OF LR INPUT FRAMES FOR 4× VIDEO SR. THE UNIT OF INPUT FRAMES AND TRAINING TIME ARE N_F AND S/BATCH. THE **TEXT** INDICATE THE BEST PERFORMANCE

Models	N_F	PSNR	Training time
DDAN-S	3	31.11	**0.364**
	5	31.16	0.728
DDAN	3	31.17	0.514
	5	**31.23**	0.941

and the number of RAB per RAG (denote as m for short). As shown in Fig. 8, there are three networks with different number of m and n, termed as DDAN-M4N4, DDAN-M6N4, DDAN-M6N6, respectively. Each network contains 4 residual blocks in the feature extraction module and 10 DCBs. We use the best model MMCNN in [41] as a reference, which has the same number of DCBs and another two deep densely residual blocks (two B5D5) as feature extraction and reconstruction respectively. We can see that larger m or n would lead to better PSNR performance. This is because the proposed network becomes deeper with larger m, n, and more hierarchical features fusion. Besides, all of our three models achieve superior PSNR performance compared with MMCNN. Therefore, we employ the DDAN-M6N6 as our best trained model DDAN.

In our experiment, we found that the ConvLSTM layer requires much larger memory costs than the convolutional layer. With large number of DCBs, the networks can face the challenge of memory footprint and the limitation of deeper architecture. To investigate the best trade-off between the DCBs and RAM. We reduce the number of DCBs and employ the same number of RAGs as DDAN to obtain another model, termed as DDAN-S. In the DDAN-S, we set the number of DCBs as 7, where the MMCNN has 10 DCBs. We compare three models DDAN-S, DDAN, and MMCNN in terms of parameters, depth and PSNR performance. As shown in Table IV. it is seen that the proposed networks combine the DCBs and RAM can achieve marked increase in term of PSNR. In particular, the model DDAN-S outperforms the MMCNN about 0.24dB with deeper layers but much fewer parameters. With the same DCBs and larger number of RAMs, our DDAN reaches much deeper framework and produce better SR results. This is because that the reduction of DCBs can obviously decrease the weight parameters and deep RAM can effectively model meaningful information to boost the reconstruction performance.

Now we explore the effectiveness of the proposed DDAN with the combinations of other different attention modules including CBAM [45], RAMSR [47], and CASR [48]. We re-implement these three attention mechanisms and conduct the same training process as our proposed RAM. The basic architecture of all models is the same as DDAN-S. Table V shows the quantitative results on *Myanmar* dataset for 4× SR. We can see that the networks with RAMSR and our RAM achieves the superior performance whereas CSAR produces SR results with slightly lower PSNR. CBAM shows the lowest PSNR than others. Then we illustrate the visual comparisons on "*city*" in *Vid4* dataset for 4× video SR. In Fig. 9, it can be observed that our proposed RAM can produce the best visual quality while CBAM and RAMSR generate the HR frame with more blurs. Though CSAR can reconstruct clear HR frame, there are some misleading contents in local regions.

G. Influence of LR Input Frames

The proposed networks can take any number of consecutive LR frames as input. In this subsection, we conduct the experiments with two different values of N_F (3 and 5) on our proposed models. In Table VI, we compare the training time of each mini-batch and the validation PSNR performance for 4× SR with different input frames 3 and 5. As shown

TABLE VII

QUANTITATIVE COMPARISONS IN TERMS OF PSNR AND SSIM OF DIFFERENT SR MODELS ON *Myanmar* TESTING DATASET WITH SCALE FACTOR 2, 3, AND 4. **TEXT** INDICATE THE BEST PERFORMANCE

Scale	Metric	Single image SR methods						
		Bicubic	A+ [3]	SRCNN [5]	VDSR [22]	DRCN [23]	LapSRN [25]	
2	PSNR	34.59	37.19	37.79	38.56	38.43	38.01	
	SSIM	0.9458	0.9638	0.9640	0.9671	0.9670	0.9656	
3	PSNR	31.59	33.48	33.88	34.64	34.71	34.57	
	SSIM	0.8957	0.9191	0.9198	0.9257	0.9262	0.9252	
4	PSNR	29.53	30.88	31.26	32.29	32.32	32.34	
	SSIM	0.8526	0.8777	0.8777	0.8873	0.8873	0.8878	
Scale	Metric	Video SR methods						
		Bayesian [6]	VSRnet [7]	MCResNet [33]	RRCN [34]	MMCNN [41]	DDAN-S (**ours**)	DDAN (**ours**)
2	PSNR	35.56	38.48	40.04	40.16	39.37	40.77	**40.84**
	SSIM	0.9515	0.9679	0.9777	**0.9794**	0.9740	0.9771	0.9775
3	PSNR	32.20	34.42	35.18	35.21	35.42	36.76	**36.85**
	SSIM	0.9203	0.9247	0.9387	0.9427	0.9393	0.9460	**0.9468**
4	PSNR	30.68	31.85	32.36	32.22	33.06	34.40	**34.46**
	SSIM	0.8895	0.8834	0.8987	0.9001	0.9040	0.9134	**0.9144**

TABLE VIII

QUANTITATIVE COMPARISONS IN TERMS OF PSNR AND SSIM OF DIFFERENT SR MODELS ON *Vid4* TESTING DATASET WITH SCALE FACTOR 2, 3, AND 4. **TEXT** INDICATE THE BEST PERFORMANCE

Scale	Metric	Single image SR methods							
		Bicubic	A+ [3]	SRCNN [5]	VDSR [22]	DRCN [23]	LapSRN [25]		
2	PSNR	28.43	30.53	30.70	31.44	31.68	31.86		
	SSIM	0.8676	0.9154	0.9172	0.9257	0.9269	0.9290		
3	PSNR	25.28	26.36	26.51	26.82	26.99	26.95		
	SSIM	0.7329	0.7904	0.7933	0.8089	0.8122	0.8158		
4	PSNR	23.79	24.59	24.69	24.98	25.03	25.06		
	SSIM	0.6332	0.6889	0.6918	0.7119	0.7141	0.7170		
Scale	Metric	Video SR methods							
		Bayesian [6]	VSRnet [7]	MCResNet [33]	DRVSR [36]	RRCN [34]	MMCNN [41]	DDRN-S (**ours**)	DDAN (**ours**)
2	PSNR	29.69	31.30	32.28	32.50	32.58	33.50	33.51	**33.65**
	SSIM	0.9055	0.9278	0.9433	0.9432	0.9451	0.9491	0.9492	**0.9517**
3	PSNR	25.82	26.79	27.54	—	27.75	28.40	28.58	**28.66**
	SSIM	0.8323	0.8098	0.8448	—	0.8560	0.8722	0.8740	**0.8752**
4	PSNR	25.06	24.84	25.45	25.90	25.54	26.28	26.37	**26.48**
	SSIM	0.7466	0.7049	0.7467	0.7678	0.7540	0.7844	0.7876	**0.7892**

in Table VI, with the increase of input frames, the models can achieve higher PSNR performance but more training time. This is because that the networks with 5 input frames can efficiently model more temporal dependencies to learn more useful information but higher processing time than 3 input frames for multi-frame SR. Meanwhile, conducting the motion compensation with more neighboring frames can involve larger computational cost, which leads to more time consuming. Therefore, we input 3 successive LR frames as input fed into our proposed networks to achieve the best trade-off between the SR performance and training time cost. In the final, we have two best trained models DDAN-S and DDAN compared with the state-of-the-arts.

H. Comparing With State-of-the-Arts

In this subsection, we conduct comprehensive comparisons of our proposed models with several single image SR methods A+ [3], SRCNN [5], VDSR [22], DRCN [23], LapSRN [25] and many state-of-the-art video SR methods including: Bayesian [6], VSRnet [7], Deep-DE [32], ESPCN [24], MCResNet [33], DRVSR [36], RRCN [34], and MMCNN [41] on 3 public video testing datasets.

1) Quantitative Comparisons: For video SR, since *Myanmar* testing dataset includes 6 scenes, each of which is composed of only 4 frames. We use 3 consecutive LR frames as input fed into our models to generate HR frames. Since that DRVSR only provide $2\times$ and $4\times$ video SR models for the fixed size 640×480 of HR frames, we do not test DRVSR on *Myanmar* dataset. As illustrated in Table VII, our proposed shallower model DDAN-S obtains higher PSNR and SSIM values for all scale factors and the deeper vision DDAN achieves the state-of-the-art. Particularly, both of our models outperforms the RRCN which adopts the *Myanmanr* as the training dataset by a considerable margin.

We further test our model on *Vid4* and *YUV21* datasets to prove the robustness of our proposed method. Since some video SR methods employ 5 consecutive frames as input to produce the center HR frame, thus, for evaluation, we skip the first and last two frames as in [7], [33]. Note that the frames from "*city*" in *Vid4* dataset are with 704×576 resolution which are not well suited for $3\times$ SR. In our experiments, we cut

TABLE IX

QUANTITATIVE COMPARISONS IN TERMS OF PSNR AND SSIM OF DIFFERENT SR MODELS ON *YUV21* WITH SCALE FACTOR 2, 3, AND 4. **TEXT** INDICATE THE BEST PERFORMANCE

Scale	Metric	Single image SR methods					
		Bicubic	A+ [3]	SRCNN [5]	VDSR [22]	DRCN [23]	LapSRN [25]
2	PSNR	30.58	33.09	33.39	34.16	34.24	34.11
	SSIM	0.8752	0.9168	0.9186	0.9266	0.9260	0.9263
3	PSNR	27.71	29.34	29.51	30.34	30.31	30.35
	SSIM	0.7727	0.8195	0.8211	0.8392	0.8378	0.8410
4	PSNR	26.29	27.50	27.66	28.39	28.27	28.45
	SSIM	0.7063	0.7499	0.7529	0.7741	0.7714	0.7780

Scale	Metric	Video SR methods						
		Bayesian [6]	VSRnet [7]	MCResNet [33]	RRCN [34]	MMCNN [41]	DDAN-S (**ours**)	DDAN (**ours**)
2	PSNR	31.99	33.54	34.37	34.68	34.96	34.81	**35.09**
	SSIM	0.8999	0.9222	0.9338	0.9378	0.9371	0.9337	**0.9408**
3	PSNR	28.62	29.58	30.11	30.32	30.82	30.92	**30.98**
	SSIM	0.8271	0.8257	0.8452	0.8530	0.8567	0.8599	**0.8608**
4	PSNR	26.14	27.64	28.08	28.16	28.90	29.13	**29.18**
	SSIM	0.7339	0.7543	0.7746	0.7785	0.7983	0.7989	**0.7990**

TABLE X

QUANTITATIVE COMPARISONS IN TERMS OF PSNR/SSIM OF DIFFERENT SR MODELS ON PUBLIC IMAGE SR DATASETS WITH SCALE FACTOR 2, 3, AND 4. **TEXT** INDICATE THE BEST PERFORMANCE

Datasets	Scale	Bicubic	A+ [3]	SRCNN [5]	VDSR [22]	DRCN [23]	LapSRN [25]	DDAN-S (**ours**)	DDAN (**ours**)
Set5	2	33.66/0.9299	36.54/0.9544	36.66/0.9542	37.53/0.9590	**37.63**/0.9588	37.52/0.9591	37.53/0.9589	37.58/**0.9593**
	3	32.58/0.9088	30.39/0.8682	32.75/0.9090	33.66/0.9213	33.82/**0.9226**	33.81/0.9220	33.76/0.9216	**33.84/0.9226**
	4	28.42/0.8104	30.28/0.8603	30.48/0.8628	31.35/0.8838	31.53/0.8854	**31.54**/0.8852	31.49/0.8848	**31.54/0.8855**
Set14	2	30.24/0.8688	32.28/0.9056	32.42/0.9063	33.03/0.9124	33.04/0.9118	32.99/**0.9124**	32.99/0.9119	**33.05/0.9124**
	3	27.55/0.7742	29.13/0.8188	29.28/0.8209	29.77/0.8314	29.76/0.8311	**29.79**/0.8325	29.71/0.8291	**29.79/0.8329**
	4	26.00/0.7027	27.32/0.7491	27.49/0.7503	28.01/0.7674	28.02/0.7670	**28.19**/0.7720	28.04/0.7669	28.12/**0.7721**
BSDS100	2	29.56/0.8431	31.21/0.8863	31.36/0.8879	**31.90/0.8960**	31.85/0.8942	31.80/0.8952	31.76/0.8945	31.82/0.8959
	3	27.21/0.7385	28.29/0.7835	28.41/0.7863	28.82/0.7976	28.80/0.7963	28.82/0.7980	28.83/0.7980	**28.88/0.7984**
	4	25.96/0.6675	26.82/0.7087	26.90/0.7101	27.29/0.7251	27.23/0.7233	27.32/0.7275	27.26/0.7237	**27.34/0.7276**
Urban100	2	26.88/0.8403	29.20/0.8938	29.50/0.8946	**30.76**/0.9140	30.75/0.9133	30.41/0.9103	30.65/0.9138	30.72/**0.9142**
	3	24.46/0.7349	26.03/0.7973	26.24/0.7989	27.14/0.8279	**27.15**/0.8276	27.07/0.8275	26.95/0.8261	**27.15/0.8279**
	4	23.14/0.6577	24.32/0.7183	24.52/0.7221	25.18/0.7524	25.14/0.7510	25.21/0.7562	25.24/0.7540	**25.33/0.7574**
Manga109	2	30.82/0.9332	35.37/0.9663	35.74/0.9661	37.22/0.9729	37.63/0.9723	37.27/0.9855	37.59/0.9726	**37.65/0.9730**
	3	26.95/0.8556	29.93/0.9089	30.48/0.9117	32.01/0.9340	32.31/0.9328	32.21/0.9318	32.33/0.9345	**32.42/0.9348**
	4	24.89/0.7866	27.03/0.8439	27.58/0.8555	28.83/0.8809	28.98/0.8816	**29.09/0.8845**	28.93/0.8816	29.01/0.8827

off the frames to 702 × 576 for 3× magnification. Similarly, with respect to the video sequences from *YUV21*, the frames in each video sequence are cut off to 351 × 288 for 3× magnification. Detailed quantitative results for the two datasets are shown in Table VIII and Table IX, respectively. It is can be seen that our DDAN-S achieves comparable performance in terms of PSNR/SSIM on all datasets with scale factor 2, 3, 4. DDAN performs better than DDAN-S, since for complex motion information, the network with deeper RAM can learn more informative features than the shallower DDAN-S for high-frequency details recovery.

To demonstrate the generalization of our proposed method, we conduct experiments on public single image SR datasets with several image SR methods. Since our proposed DDAN-S and DDAN both take 3 adjacent frames as input for SR reconstruction, for each image from image SR datasets, we repeat the image twice and obtain three same images as a video clip. As shown in Table X, though the proposed DDAN and DDAN-S are not trained on image SR datasets, our models still achieve competitive image SR results across all datasets and scales.

2) Qualitative Comparisons: In addition to the quantitative evaluation, we shows the visual comparisons of the different SR methods for 4× SR in Fig. 10, and Fig. 11. Since that the MMCNN only provide the original training code without pretrained models, we retrain the best models introduced in the paper to obtain the subjective results. The results of ESPCN [24] are cited from the public results in VESPCN [35]. In Fig. 10, we can see that our models can produce clearer lines, and shaper edges, while other methods are prone to produce lines with more blurs. Moreover, as sketched in Fig.11, the parts including letters or numbers in the calendar are magnified for more obvious comparison. It is observed that after 4× magnification by Deep-DE, ESPCN, VDSR, and LapSRN, the numbers can still be identified while the letters are hard to be recognized. DRVSR has the ability to recover part information of the letters, but still produces local details with poor quality. Although the proposed DDAN-S produces the HR frame with lower PSNR and SSIM values, the model can reconstruct HR image with sharper and clearer characters.

3) Super-Resolving Real-World Video Sequences: To further demonstrate the effectiveness of our propose method,

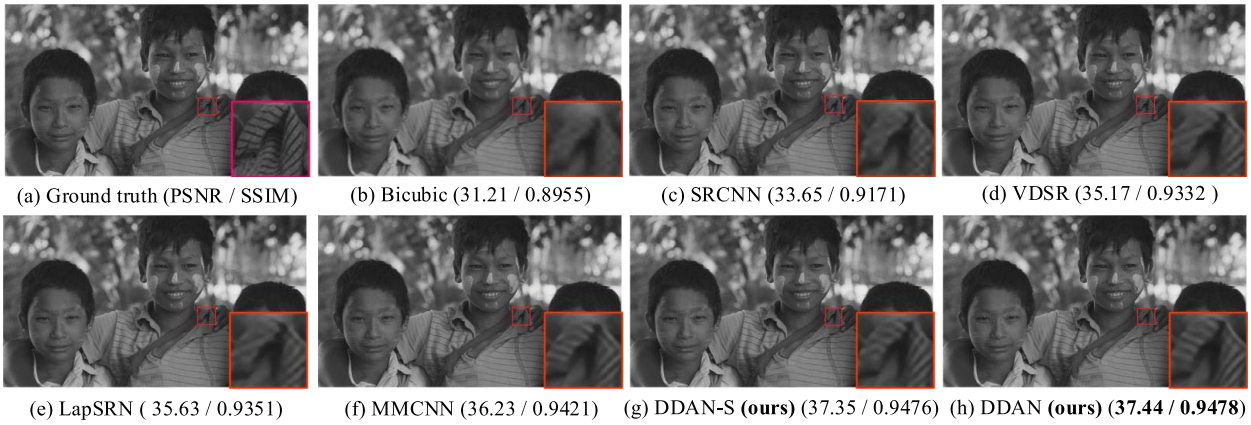


Fig. 10. Visual comparisons on the *Myanmar* testing dataset, where upscaling factor is 4.

Fig. 11. Visual comparisons on the 15^{th} frame from *calendar* for 4× SR.

we capture two real-world LR video clips as shown in Fig. 12. For the two examples, neither the ground-truth videos nor the downsampling kernels are available. We extract 31 consecutive frames from each videos and compare the 15^{th} frame with other video SR methods. In Fig. 12, we can observe that both of our two models can produce the SR results with clearer letters, numbers, and more photo-realistic details than the most state-of-the art method MMCNN. Although Deep-DE can produce clearer characters in some parts, the images contains much more significant artifacts and blurs than our results.

4) Inference Time: As for inference time, for fair comparison, we use the public codes of the compared algorithms to evaluate the runtime on the machine with 3.4 GHz Intel i7 CPU (128G RAM) and 1 NVIDIA Titan Xp GPU (12G Memory). The average running time and PSNR values of different methods on *Vid4* dataset for 4× SR are shown in Table XI. Besides, we compare the model complexity of all SR methods in terms of parameter amount which are illustrated in Table XI. As we can see, SRCNN and VSRnet have fewer parameters but achieve worse SR performance with much slower reconstruction speed compared with other methods. Though DRVSR produces HR frames with the fastest speed, this method still generates the SR results with lower PSNR performance than MMCNN and our models. The proposed

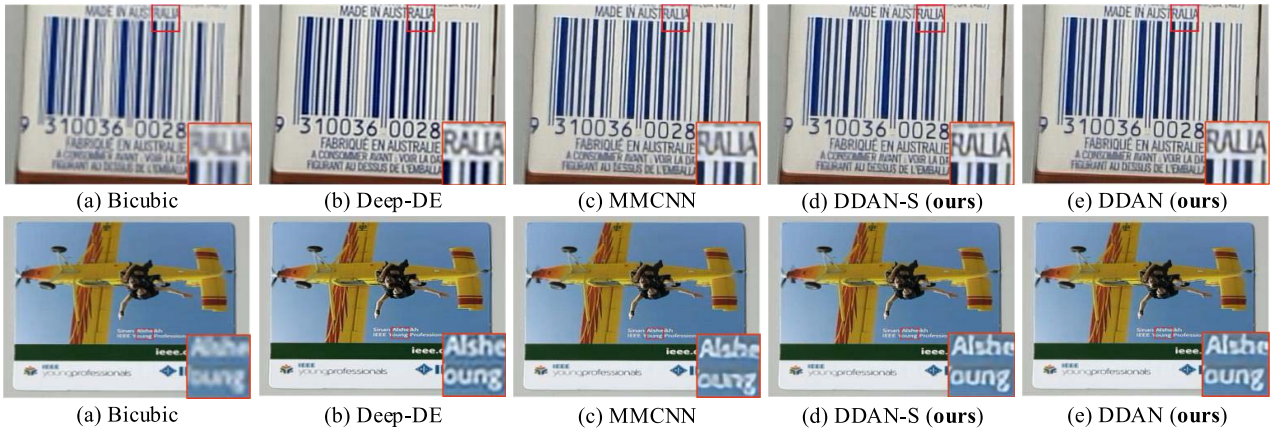


Fig. 12. Visual comparisons on real-world LR videos by 4× magnification. The original LR video clips are at the resolution of 116 × 83 (top) and 199 × 218 (bottom).

TABLE XI

COMPARISONS ON PSNR/SSIM VALUES, MODEL PARAMETERS, AND INFERENCE TIME ON *Vid4* DATATSET FOR 4× SR. **TEXT** INDICATE THE BEST PERFORMANCE

Methods	SRCNN [5]	VDSR [22]	LapSRN [25]	VSRnet [7]	DRVSR [36]	MMCNN [41]	DDAN-S (**ours**)	DDAN (**ours**)
PSNR	24.69	24.98	25.06	24.84	25.90	26.28	26.37	**26.48**
SSIM	0.6918	0.7119	0.7170	0.7049	0.7678	0.7844	0.7876	**0.7892**
Time (sec)	10.011	0.077	1.183	6.132	**0.053**	0.201	0.187	0.216
Parameters	57K	665K	813K	78K	1.722M	10.582M	7.051M	10.290M

DDAN-S can achieve superior PSNR/SSIM values with faster reconstruction speed than MMCNN. Moreover, our best model DDAN can obtain the highest quantitative performance with slightly higher time cost than DDAN-S.

VI. CONCLUSION

In this paper, we have proposed a deep dual attention network for video SR. Our model investigates multi-level optical flow representations between the adjacent frames and center frame in a coarse-to-fine manner and infers the spatial transform to model the motion compensation. We extract the detail components of neighboring frames and employ dual attention mechanisms to make full use of spatio-temporal meaningful information for more accurate HR videos reconstruction. We compare our models with other recent state-of-the-art video SR approaches and the results demonstrate that our proposed method obtains superior SR performance on public benchmark datasets.

REFERENCES

[1] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 349–356.

[2] R. Timofe, V. D. Smet, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1920–1927.

[3] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asia Conf. Comput. Vis.*, 2014, pp. 111–126.

[4] Z. Zhu, F. Guo, H. Yu, and C. Chen, "Fast single image super-resolution via self-example learning and sparse representation," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2178–2190, Dec. 2014.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[6] 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.

[7] 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.

[8] A. Lucas, S. L. Tapia, R. Molina, and A. K. Katsaggelos, "Generative adversarial networks and perceptual losses for video super-resolution," 2018, *arXiv:1806.05764*. [Online]. Available: https://arxiv.org/abs/1806.05764

[9] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super resolution," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1327–1344, Sep. 2004.

[10] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1958–1975, Sep. 2009.

[11] S. P. Belekos, N. P. Galatsanos, and A. K. Katsaggelos, "Maximum *a posteriori* video super-resolution using a new multichannel image prior," *IEEE Trans. Image. Process.*, vol. 19, no. 6, pp. 1451–1464, Feb. 2010.

[12] X. Li and M. T. Orchard, "New edge directed interpolation," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Sep. 2010, pp. 311–314.

[13] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.

[14] X. Liu, D. Zhao, J. Zhou, W. Gao, and H. Sun, "Image interpolation via graph-based Bayesian label propagation," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1084–1096, Mar. 2014.

[15] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 49–64.

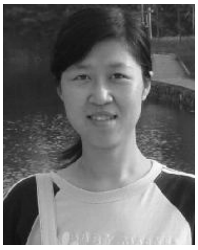
[16] J. Yang, Z. Lin, and S. Cohen, "Fast image super-resolution based on in-place example regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1059–1066.

[17] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5197–5206.

[18] K. Jia, X. Wang, and X. Tang, "Image transformation based on learning dictionaries across image spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 367–380, Feb. 2013.

[19] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3791–3799.
[20] Y. Zhang, Y. Zhang, J. Zhang, and Q. Dai, "CCR: Clustering and collaborative representation for fast single image super-resolution," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 405–417, Mar. 2016.
[21] Y. Tang and L. Shao, "Pairwise operator learning for patch-based single-image super-resolution," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 994–1003, Dec. 2017.
[22] 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.*, Jun. 2015, pp. 1646–1654.
[23] 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.*, Jun. 2016, pp. 1637–1645.
[24] W. Shi *et al.*, "Real-time single image and video superresolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1874–1883.
[25] 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.*, Jul. 2017, pp. 5835–5843.
[26] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2790–2798.
[27] Z. Ma, 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 Recognit.*, Jun. 2015, pp. 5224–5232.
[28] Q. Dai, S. Yoo, A. Kappeler, and A. K. Katsaggelos, "Sparse representation based multiple frame video super-resolution," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 765–781, Nov. 2017.
[29] D. Mitzel, T. Pock, T. Schoenemann, and D. Cremers, "Video super resolution using duality based TV-L-1 optical flow," in *Proc. 31st DAGM Symp. Pattern Recognit.*, 2009, pp. 432–441.
[30] Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," in *Proc. Adv. Neural. Inf. Process.*, vol. 2015, pp. 235–243.
[31] J. Guo and H. Chao, "Building an end-to-end spatial-temporal convolutional network for video super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4053–4060.
[32] 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.*, Dec. 2015, pp. 531–539.
[33] D. Li and Z. Wang, "Video superresolution via motion compensation and deep residual learning," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 749–762, Dec. 2017.
[34] D. Li, Y. Liu, and Z. Wang, "Video super-resolution using non-simultaneous fully recurrent convolutional network," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1342–1355, Mar. 2019.
[35] J. Caballero *et al.*, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2848–2857.
[36] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4482–4490.
[37] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6626–6634.
[38] D. Liu *et al.*, "Robust video super-resolution with learned temporal dynamics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2526–2534.
[39] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.
[40] X. Shi, Z. Chen, H. Wang, and D. Y. Yeung, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural. Inf. Process.*, 2015, pp. 802–810.
[41] Z. Wang *et al.*, "Multi-memory convolutional neural network for video super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2530–2544, Dec. 2019.
[42] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural. Inf. Process.*, 2017, pp. 6000–6010.
[43] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3156–3164.
[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[45] W. Sanghyun, P. Jongchan, L. J. Young, and S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Euro. Conf. Comput. Vis.*, 2018, pp. 3–19.
[46] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 294–310.
[47] J. H. Kim, J. H. Choi, M. Cheon, and J. S. Lee, "RAM: Residual attention module for single image super-resolution," 2018, *arXiv:1811.12043*. [Online]. Available: https://arxiv.org/abs/1811.12043
[48] Y. Hu, J. Li, Y. Huang, and X. Gao, "Channel-wise and spatial feature modulation network for single image super-resolution," 2018, *arXiv:1809.11130*. [Online]. Available: https://arxiv.org/abs/1809.11130
[49] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural. Inf. Process.*, 2015, pp. 2017–2025.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*. [Online]. Available: https://arxiv.org/abs/1512.03385
[51] 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.*, Dec. 2015, pp. 1026–1034.
[52] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 135-1–135-10.
[53] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.*, 2012, pp. 711–730.
[54] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
[55] Y. Matsui *et al.*, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21811–21838, 2017.
[56] X. Lorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.
[57] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

Feng Li received the B.S. degree from Anhui Normal University, China, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Information Science, Beijing Jiaotong University, Beijing, China. His research interests are in image and video compression, image and video super resolution, and other low-level computer vision tasks, with the focus on deep-learning-based methods.



Huihui Bai received the B.S. and Ph.D. degrees from Beijing Jiaotong University, China, in 2001 and 2008, respectively. She is currently a Professor with Beijing Jiaotong University. She has been engaged in research and development work in video coding technologies and standards, such as HEVC, 3D video compression, multiple description video coding (MDC), and distributed video coding (DVC).

Yao Zhao (Senior Member, IEEE) received the B.S. degree from Radio Engineering Department, Fuzhou University, China, in 1989, the M.E. degree from Radio Engineering Department, Southeast University, Nanjing, China, in 1992, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), China, in 1996. He is currently the Director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He serves on the Editorial Board of several international journals, including as an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS and the IEEE SIGNAL PROCESSING LETTERS and an Area Editor for *Signal Processing: Image Communication* (Elsevier). He was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013. He was named as a Distinguished Young Scholar by the National Science Foundation of China in 2010.