Residual Invertible Spatio-Temporal Network for Video Super-Resolution

Xiaobin Zhu,[1,2*] **Zhuangzi Li,**[2*] **Xiao-Yu Zhang,**[3†] **Changsheng Li,**[4†] **Yaqi Liu,**[3] **Ziyu Xue**[5]
[1]School of Computer and Communication Engineering, University of Science and Technology Beijing
[2]School of Computer and Information Engineering, Beijing Technology and Business University
[3]Institute of Information Engineering, Chinese Academy of Sciences
[4]University of Electronic Science and Technology of China
[5]Information Technology Institute, Academy of Broadcasting Science, NRTA, China
{brucezhucas, lizhuangzi}@gmail.com, zhangxiaoyu@iie.ac.cn,
lichangsheng@uestc.edu.cn, liuyaqi@iie.ac.cn, xueziyu@abs.ac.cn

Abstract

Video super-resolution is a challenging task, which has attracted great attention in research and industry communities. In this paper, we propose a novel end-to-end architecture, called Residual Invertible Spatio-Temporal Network (RISTN) for video super-resolution. The RISTN can sufficiently exploit the spatial information from low-resolution to high-resolution, and effectively models the temporal consistency from consecutive video frames. Compared with existing recurrent convolutional network based approaches, RISTN is much deeper but more efficient. It consists of three major components: In the spatial component, a lightweight residual invertible block is designed to reduce information loss during feature transformation and provide robust feature representations. In the temporal component, a novel recurrent convolutional model with residual dense connections is proposed to construct deeper network and avoid feature degradation. In the reconstruction component, a new fusion method based on the sparse strategy is proposed to integrate the spatial and temporal features. Experiments on public benchmark datasets demonstrate that RISTN outperforms the state-of-the-art methods.

Introduction

Video super-resolution (VSR) aims to generate high-resolution (HR) video frames from its low-resolution (LR) version, as shown in Figure 1. It can be widely applied to various intelligent image processing tasks, e.g., satellite videos (Demirel and Anbarjafari 2011), the surveillance and 4K televisions recovery (Zhang et al. 2010), and so on. VSR is a long-standing challenging task, mainly due to the following two reasons: Firstly, super-resolution is an inherently ill-posed problem for its one-to-many mapping nature, i.e., one LR frame can map to various HR frames. Secondly, there is no a satisfying architecture designed to integrate spatial and temporal information in a joint framework by far.

Along with flourishing of Convolutional Neural Networks (CNNs), a series of single image super-resolution approaches emerged and achieved promising performances (Dong et al. 2016; Tong et al. 2017; Zhang et al. 2018). However, the single-image methods completely ignore the intrinsic temporal information and motion nature, therefore can not well adapted to the VSR task. Some deep architectures were proposed to capture temporal consistency of consecutive frames, which can be mainly divided into two categories: motion compensation based approaches (Kappeler et al. 2016; Caballero et al. 2017; Tao et al. 2017), and Recurrent Convolutional Networks (RCNs) based approaches (Guo and Chao 2017; Huang, Wang, and Wang 2018; Yang et al. 2018; Liu et al. 2018). The motion compensation based approaches tried to extract explicit motion information, e.g., optical flow, to model the temporal dependency for highlighting visual continuity. However, the high computational cost of these type of approaches is terrifying. Afterwards, the RCNs based approaches were proposed to extract the motion information of consecutive frames and enhance visual quality in an end-to-end manner. For instance, in (Guo and Chao 2017), the stack of convolutional layers was adopted to extract the spatial and content information of a input single frame, and the recurrent convolutional layers is adopted to capture the temporal information of consecutive frames.

The aforementioned RCNs based approaches can adaptively deal with complicated and large motions with relatively low computational cost. However, the performances are always not remarkable for the following reasons: 1) The LR video frames must be interpolated in advance, which not only consumes a large amount of additional memory space,

Bicubic Ours HR

Figure 1: Side-by-side comparisons of bicubic interpolation, our result, and HR ground truth for $4\times$ upsampling.

*These authors contributed equally to this study and share the first authorship.
†Corresponding authors.

but also hinders the construction of much deeper network. 2) The existing methods can not well keep spatial information. The spatial information makes sure the input LR video frames and the corresponding super-resolved frames should have more structural similarity. 3) The majority of the existing recurrent architectures is not deep enough to effectively cover the long-range motion and temporal consistency in VSR. 4) Spatial features and temporal features are not integrated for boosting the performance.

In this paper, a novel efficient Residual Invertible Spatio-Temporal Network (RISTN) is proposed to tackle the abovementioned problems, which mainly consists of three components, namely the spatial component, the temporal component and the reconstruction component. In the spatial component, Residual Invertible Block (RIB) is designed to extract informative features with spatial information. In the temporal component, a residual dense convolutional LSTM (RDC-LSTM) is presented to learn sequential feature representation. Finally, a reconstruction component is adopted to integrate spatial features and temporal features into a joint framework.

In summary, the main contributions of this paper can be concluded as follows:

- We propose a novel Residual Invertible Spatio-Temporal Network (RISTN) with much deeper structure compared with existing recurrent convolutional networks based approaches, for achieving high model accuracy in an efficient way.
- Inspired by the algorithm of Invertible Block (Jacobsen, Smeulders, and Oyallon 2018), a lightweight Residual Invertible Block (RIB) is designed to better keep the spatial information between the LR video frames and the corresponding super-revolved frames. In RIB, the residual connection is introduced to learn fine-grained feature representation, meanwhile degrade the information loss.
- A novel residual dense convolutional LSTM (RDC-LSTM) is presented. It can not only capture the temporal information of consecutive video frames, but also effectively transform spatial features from different hierarchical levels.
- We present a sparse features fusion strategy for combining spatial and temporal features to reconstruct the final output. The sparse features fusion can select informative features and model the mapping across the low-quality and high-quality video frames in an adaptive way.
- Extensive experiments conducted on benchmark datasets demonstrate the effectiveness and efficiency of the proposed RISTN compared with the state-of-the-arts.

Related Work

In this section, some related works are introduced from three aspects: Firstly, we will introduce some single image super-resolution works. Secondly, conventional technologies of VSR are illuminated. Thirdly, two branches of deep learning methods will be comparatively illustrated.

Single image super-resolution aims to generate a HR image from it's corresponding low-resolution LR image. The seminal CNNs based approach was proposed in (Dong et al. 2016). Later, many works focused on the improvement of CNNs architecture. On the one hand, some of approaches proposed deeper architectures with shortcut connections and achieved fruitful experimental results (Kim, Lee, and Lee 2016a; Tong et al. 2017; Zhang et al. 2018). On the other hand, some of works aimed to design more simplified and efficient model using recursive architectures (Kim, Lee, and Lee 2016b; Tai, Yang, and Liu 2017; Han et al. 2018). All those improvements tremendously promote the development of super-resolution task.

VSR is an expand of single image super-resolution, and mainly investigates two categories of traces: the intra-frame spatial relationship and the inter-frame temporal relationship. Conventional approaches extracted optical flow for motion estimation to compensate temporal information (Fransens, Strecha, and Gool 2004; Mitzel et al. 2009). In (Liu and Sun 2011), a Bayesian framework was proposed to estimate HR video sequences, in which they simultaneously computed the motion fields and blur kernels. Ma et al. presented an algorithm that extended the same idea to handle motion blur (Ma et al. 2015). However, these methods all suffered from the computation cost. Despite they constructed complex motion compensation models, the constructed VSR models themselves was difficult to learn the delicate compensated information.

Deep convolutional networks for VSR have achieved promising results in recent works, which can be divided into two categories: The one is motion compensation algorithms based approaches, the other is recurrent convolutional networks based approaches (RCNs). In motion compensation algorithms based approaches, Kappeler et al. proposed a general framework in which the motion compensation algorithm and the CNNs were combined (Kappeler et al. 2016). In (Caballero et al. 2017), The 3D convolution and sub-pixel convolution were adopted to improve the efficiency of the learning process. In (Tao et al. 2017), a sub-pixel motion compensation layer was proposed and combined with a recurrent convolutional network. And Sajjadi et al. proposed an end-to-end trainable frame-recurrent video super-resolution framework that adopted the previously inferred HR estimation to super-resolve the subsequent frame in (Sajjadi, Vemulapalli, and Brown 2018). However, there are still much time consumptions and computational loads in the motion compensation process.

The RCNs based approaches directly build an end-to-end network without explicit motion compensation and also show the superiority in complex motions reconstruction. In (Guo and Chao 2017), a spatio-temporal network for VSR was proposed, who built a infrastructural architecture of RCNs. In (Liu et al. 2018), Liu et al. effectively utilized temporal information by a temporal adaptive neural network and a spatial alignment network. And some works directly built a light architecture to achieve high efficiency (Huang, Wang, and Wang 2018; Yang et al. 2018). While they are difficult to achieve satisfied quality for their shallow architectures and incomplete feature representations. Our work aims to improve the RCNs and enhance the transformation capabilities on the basis of some the state-of-the-art network architec-

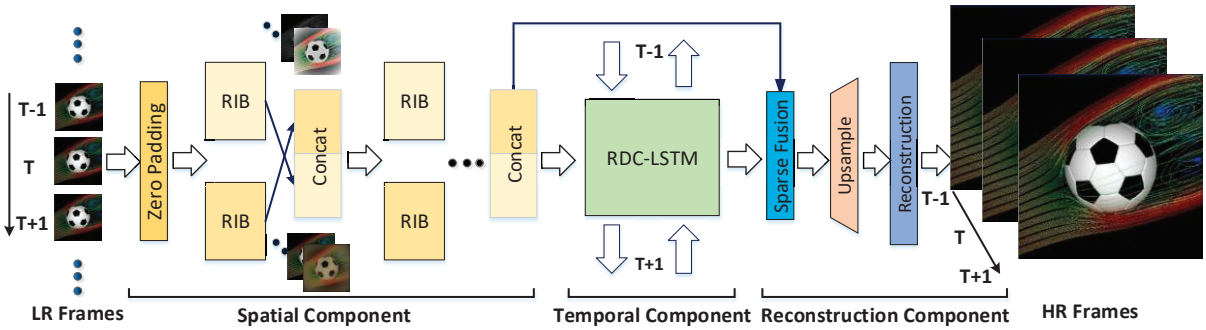


Figure 2: The framework of the proposed residual invertible spatio-temporal network (RISTN).

tures (He et al. 2016; Huang et al. 2017; Gomez et al. 2017; Jacobsen, Smeulders, and Oyallon 2018).

Our Method

Overview

The framework of the proposed RISTN is shown in Figure 2. In the spatial component, the sequential LR frames are feed into a padding layer, which builds an initial feature maps by zero padding on RGB channels. The two followed parallel residual invertible blocks (RIBs) have different architectures with different numbers of layers for exploiting hierarchical features. The output feature maps of the former RIB will be concatenated and then put into the next parallel RIBs. Notably, the concat operations of concatenation can effectively increase the diversities of feature maps. In the temporal component, a residual dense convolutional Long Short-Term Memory (RDC-LSTM) network is proposed to handle features of continuous frames. In the reconstruction component, a sparse feature fusion method is proposed to integrate the spatial and temporal feature maps, the fused feature maps upsampled to the target HR size. Finally, the reconstruction layer is adopted to recover the RGB-channel HR frames.

The ultimate goal of VSR is to train a generating function F that estimates the HR frames while given the LR frames. Given the current low-resolution LR frames I_T^{LR} and corresponding ground-truth HR frames I^{HR}, the VSR can be formulated as:

$$I_T^{HR} = F(\{I_T^{LR}, I_{T+i}^{LR}\}),\ i \in \{\pm 1, ..., \pm k\}, \qquad (1)$$

where the $|$ denotes the conditional probability, T represents the current timestamp. i denotes the consecutive i-th timestamp. In our work F can be represented by the proposed RISTN.

Residual invertible block

The super-resolved frames should have similar structures with the input LR frames, and this important property is called the spatial information. However, previous works can not sufficiently fulfill the spatial information for the lossy features they use (Huang, Wang, and Wang 2018; Yang et al. 2018). In the algorithm of invertible block (Jacobsen, Smeulders, and Oyallon 2018), it retains all information about the input signals in any of their intermediate representations. In other words, the spatial information can be maintained for its lossless feature transformation process. However the invertible nature of the invertible block (IB) limits its ability to learn rich reconstruction features for the ill-posed super-resolution task. Inspired by (He et al. 2016), we propose a residual invertible block (RIB), in which the residual connection is constructed and the parallel invertible block is designed to learn the difference between LR and HR frames.

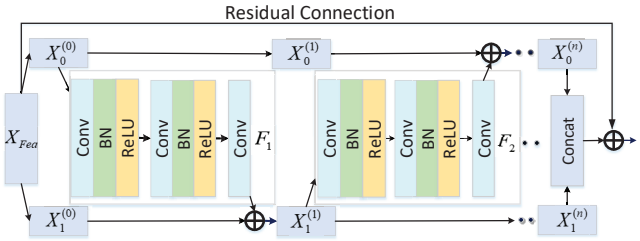


Figure 3: The residual invertible block architecture, $\oplus$ denotes the element-wise addition.

As shown in Figure 3, given the input features X_{Fea}, it is splited into two sublayers, the $X_0^{(0)}$ and $X_1^{(0)}$. We define $F_i, i \in [1, 2, ..., n-1]$ called convolutional bottleneck. The convolutional bottleneck consists of convolutional layers, Batch Normalizations (BNs) and Rectified Linear Units (ReLUs). The features $X_1^{(1)}$ can be calculated as:

$$X_1^{(1)} = X_1^{(0)} + F_1(X_0^{(0)}). \qquad (2)$$

On the contrary, $X_1^{(0)}$ can be inversely represented by $X_1^{(1)} - F_1(X_0^{(0)})$. Therefore, an inference can be reached: Given features from the i-th layer $(X_0^{(i)}, X_1^{(i)})$, their previous feature representation $X_1^{(i-1)}$ and $X_0^{(i-1)}$ are:

$$X_1^{(i-1)} = X_1^{(i)} - F_i(X_0^{(i-1)}), \qquad (3)$$

$$X_0^{(i-1)} = X_0^{(i)}. \qquad (4)$$

According to above formulas, previous features can be sequentially represented from any $X_1^{(i)}$ and $X_0^{(i)}$. Besides, the output of a RIB can be written as:

$$X_{out} = [X_0^{(n)}, X_1^{(n)}] + X_{Fea}, \qquad (5)$$

X_{Fea} is the input of RIB. $[,]$ denotes the concatenation of feature maps. It can be seen that RIB is a memory-efficient structure, that only half feature maps are calculated after each convolutional bottleneck (Gomez et al. 2017). According to Formula (5), it can be seen that the concatenated feature maps generated by IB try to approximate the difference between the input X_{Fea} and the target output feature maps, thus the IBs in the consecutive RIBs can learn the difference between LR and HR frames.

Recurrent model with shortcut connections

In this component, the convolutional LSTM (C-LSTM) is adopted to dig out informative features of consecutive frames. Different from conventional one-dimensional LSTM, C-LSTM captures 2D features from neighbouring timestamps. For thoroughly exploiting temporal consistency, the C-LSTM is constructed as a bidirectional architecture. We build the bidirectional LSTM cell as (Graves, Jaitly, and Mohamed 2013) that outputs of forward and backward on timer shaft are concatenated as the output of one cell. The schematic diagram of different C-LSTM architectures are shown in Figure 4.

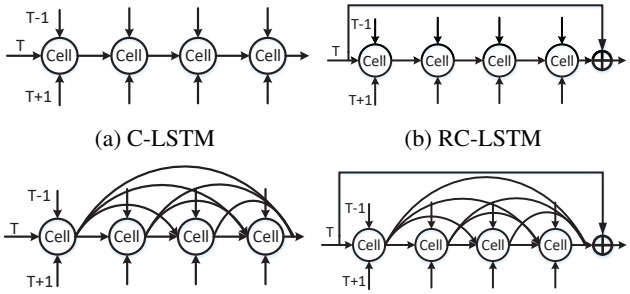


Figure 4: C-LSTM architectures, (a) is the original version, (b), (c) and (d) are the proposed shortcut connected structures. $\oplus$ denotes element-wise addition.

There are still some shortcomings for the original C-LSTM. With the increase of the network depth, the performance of the original C-LSTM can not be accordingly promoted. Furthermore, the vanishing gradient problem is also serious. Thus, three variations with shortcut connections are proposed for better feature transformation, residual connected LSTM (RC-LSTM) is shown in Figure 4 (b), which aims to optimize the residual value of input and output. It is good at learning informative diversities and solving gradient problems. Dense connected LSTM (DC-LSTM) is shown in Figure 4 (c), which exploit hierarchical features from different cell levels and alleviate the vanishing-gradient problem. the final model residual dense connected LSTM (RDC-LSTM) is shown in Figure 4 (d), it combines the advantages of residual connections and dense connections for both supplement each other. The effectiveness of shortcut connections are also verified in experiments. Notably, the Figure 4 (b) and Figure 4 (d) have the element-wise addition, which must guarantee equivalent channels. Therefore, we introduce an auxiliary convolutional layer to convert the feature maps received from spatial component, it aims to keep the same channels between input and output of the recurrent model. An example of Figure 4 (d) can be represented as:

$$X_{out} = W_{1\times1\times c\times c'} * X_{in} + [H_0, H_1, ..., H_{n-1}]_{c'}, \quad (6)$$

where $[H_0, H_1, ..., H_{n-1}]$ denotes the concatenation of the feature maps produced in all preceding layers, X_{in}, X_{out} is the input and output of RDC-LSTM. W is convolutional filter matrix that the kernel size is 1×1. c is input the channel number, "$*$" represents the convolution operation, converting c to $c^{'}$.

Sparse feature fusion

Previous works ignore the combination of spatial and temporal features in the reconstruction component (Guo and Chao 2017; Yang et al. 2018). In this section, we fuse the spatial and temporal features using a sparse strategy. In my opinion, the original spatial information should be taken into consideration in the final reconstruction due to features degradation caused by the consecutive LSTM layers. The sparsity is adopted to select useful feature maps for reducing the risk of overfitting (Rubinstein, Zibulevsky, and Elad 2010). The flowchart of the proposed sparse feature fusion method is shown in Figure 5.

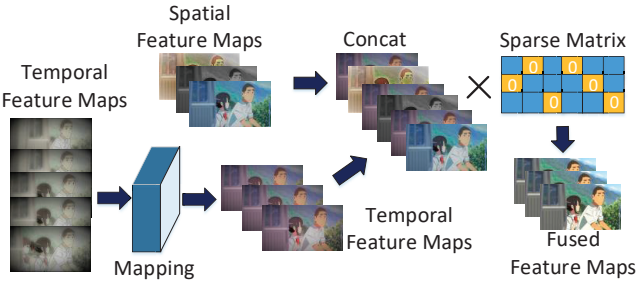


Figure 5: The flowchart of the sparse feature fusion.

The temporal features will be transformed into the same space with the spatial features using a mapping layer. Assuming that the spatial feature maps X_s have $c1$ channels, temporal feature maps X_t have $c2$ channels. We define $c = 2 \times c1$, the concatenated feature maps X_{concat} can be represented as:

$$X_{concat} = [W_{1\times1\times c2\times c1} * X_t, X_s]_c, \quad (7)$$

where W is the convolutional filter of temporal-to-spatial mapping. $c2$ is the input numbers of channel and $c1$ is the output numbers of channel. "$*$" represents the convolution operation. $[,]$ is the cross concatenation. Then, a sparse matrix $SM \in \mathbb{R}^{c\times c/2}$, is designed to select useful feature maps and compress feature channels in an adaptive way. The fused feature maps X_{fused} can be calculated as:

$$X_{fused} = X_{concat} \times SM, \quad (8)$$

where "$\times$" denotes matrix multiplication. In addition, the sparsity of SM is controlled by a L1 regular term in the training loss.

Upsampling during reconstruction

In the reconstruction component, deconvolutional layers are constructed to upsample the feature maps to the resolution of HR. In previous RCNs based methods (Guo and Chao 2017; Huang, Wang, and Wang 2018; Yang et al. 2018), bicubic interpolation is used to upsample LR frames initially, and the upsampled frames are put into the networks. It increases the computational complexity for SR. In addition, interpolation approaches are uninformative for solving the SR problem. Inspired by progress of single image super-resolution (Tong et al. 2017), we employ deconvolution layers as the upsampling layer in reconstruction component that the transformed features are upsampled at the end of the network. Different from sub-pixel convolutions (Caballero et al. 2017), deconvolution layer adaptively allows arbitrary channel numbers as input rather than the fixed numbers. In our work, two stacks of deconvolution layers with small 3×3 kernels and 256 feature maps are adopted for upsampling feature maps.

Training and loss

We firstly pre-train our spatial network on ImageNet dataset, in which the pixel-wise mean square error (MSE) loss is adopted as the loss metric. The MSE loss can be formulated as:

$$L_{pre} = ||I^{HR} - F(I^{LR})||_2^2, \tag{9}$$

where the I^{LR} denotes the input low-resolution patch, and I^{HR} is ground truth high-resolution patch. $||.||_2$ denotes the $L2$ norm. Then, the RISTN is trained on the video dataset and the training loss L can be represented as:

$$L = \sum_{T=-k}^{k} ||I_T^{HR} - F(I_T^{LR})||_2^2 + \lambda ||SM||_1, \tag{10}$$

where k is the total number of consecutive frames, $T = 0$ represents the current frame, λ is the hyper-parameter set by users, and SM represents the sparse matrix of the fusion part. The $L1$ norm can make sure the sparsity of SM (Rubinstein, Zibulevsky, and Elad 2010).

Experimental Evaluation

Dataset and metrics

In our approach, the randomly selected 50,000 images from ImageNet are adopted for the spatial network pre-training. And 195 videos of 1080p (1920 $\times$1080) are collected from 699pic.com and vimeo.com, which include different scenarios: nature, streetscape and daily life, etc.. The collected videos are downscaled by times 2 (960 $\times$540) and randomly clipped to 5800 video sequences, which all consists of consecutive video frames. With no loss of generality, 5 consecutive video frames in each sequence are used for training. The public available benchmark dataset of Vide4 (Liu and Sun 2011) is used to demonstrate the performance of the RISTN. All experiments are performed using $4\times$ upscaling factors from low resolution to high resolution. The peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) are evaluation criterion. According to the state-of-the-art approaches, the PSNR and SSIM are all calculated on the individual Y- channel.

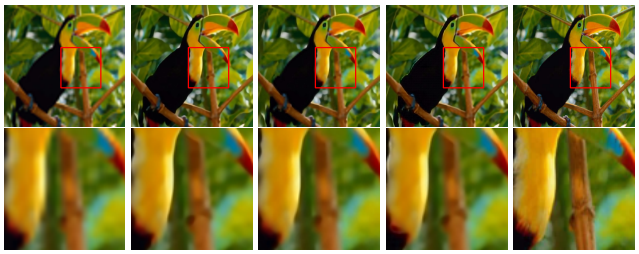
RIB evaluation

To demonstrate the effectiveness of RIB, we re-architect the RISTN, in which the temporal component is removed. The bicubic upsampling is introduced as a baseline for comparison, and is denoted as Bic. Two state-of-the-art image super-resolution methods, DRCN (Kim, Lee, and Lee 2016b), DSRN (Han et al. 2018) and RDN (Zhang et al. 2018) are adopted for comparisons. Invertible block (IB), which abandons the residual connection and residual block (RB), which only utilize residual connection without invertible structure based CNN are also compared to testify the effectiveness and necessity of the RIB architecture. The size of the LR patch is set as 30×30 and the ground truth HR patch is set as 120×120.

Table 1: The evaluation of RIB ($4\times$) on the Set5 dataset.

Method	Bic	DRCN	DSRN	RDN	IB	RB	RIB
PSNR	28.42	31.53	31.40	32.47	31.30	31.49	31.65
SSIM	0.810	0.884	0.883	0.899	0.872	0.885	0.897
Params	0M	1.75M	1.25M	22M	1.21M	6.08M	1.21M

As shown in Table 1, RIB achieves the highest scores among all the compared methods. By adopting the residual architecture, the PSNR score of RIB rises by 0.35 dB to 31.65 dB than IB. The RIB also outperforms RB by 0.16 dB of PSNR. Compared with the state-of-the-art DRCN and DSRN, RIB also surpasses them more than 0.12 dB and 0.25 dB in PSNR. Meanwhile, RIB has less parameters, which reduce DRCN and DSRN by 0.54 M and 0.04 M, due to the memory-efficient structure (Gomez et al. 2017). All the experiments can demonstrate the effectiveness and necessity of RIB. Though RDN achieves promising results, the parameter scale of RDN (about 22M) is nearly 20 times of our model. Visual comparisons are shown in Figure 6, relatively clear feathers are recovered by RIB.



Bic DRCN IB RIB HR

Figure 6: The visual comparisons of RIB evaluation ($4\times$).

Recurrent architecture evaluation

Three kinds of recurrent model with shortcut connections are evaluated in this section, namely RC-LSTM, DC-LSTM and RDC-LSTM (Readers can kindly refer to Figure 4). The C-LSTM is viewed as the baseline for comparisons. For each recurrent model, it is testified with different numbers of layers. In other words, the 3-layer, 5-layer, 8-layer, 10-layer models are all testified and compared. The input LR patch

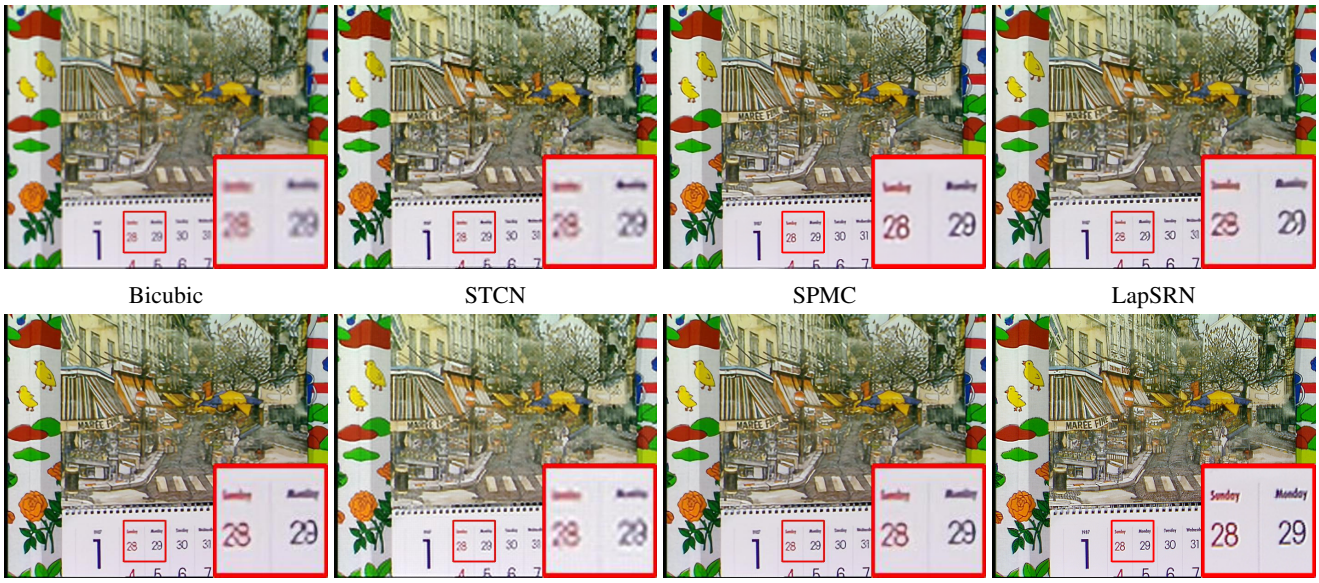


Figure 7: The visual comparisons on "Calendar" of Vid4 with scaling factors as $4\times$.

Table 2: Comparison of average PSNR and SSIM on Vid4 dataset for scaling factor 4.

Method	Bic	VSRNet	STCN	VESPCN	SPMC	LapSRN	BRCN	RDN	Liu et al.	RISTN-NF	RISTN-DF	RISTN
PSNR	23.74	24.41	24.91	25.35	25.52	25.15	24.43	25.30	25.88	25.74	25.97	**26.13**
SSIM	0.633	0.707	0.734	0.757	0.760	0.771	0.712	0.750	0.767	0.782	0.789	**0.792**

size is set as 40×40, and the ground-truth HR patch is set as 160×160.

Table 3: The PSNR of recurrent model ($4\times$) on Vid4 dataset with different layer numbers.

Layers \ Method	C-LSTM	RC-LSTM	DC-LSTM	RDC-LSTM
3	24.81	**25.18**	25.09	25.16
5	24.90	25.26	25.13	**25.35**
8	21.98	25.35	25.29	**25.47**
10	22.13	25.40	25.36	**25.60**

As shown in Table 3, when the layer number is set as 3, RC-LSTM achieves the highest score, while the PSNR of RDC-LSTM is slightly lower than RC-LSTM. It can be explained that dense connections have less feature maps in a shallow architecture, which is not conducive to exploit informative features. On the other hand, due to the vanishing gradient problem, the C-LSTM structure will decrease the when the layer number increases. The other recurrent architectures with shortcut connections produce better performance with increasing layer numbers. Obviously, the RDC-LSTM can achieve more prominent advantages with deeper layers. As shown in Table 3, the RDC-LSTM outperforms RC-LSTM and DC-LSTM by 0.13 dB and 0.17 dB of PSNR in the condition of 10 layers. It can be concluded that RDC-LSTM can effectively combine the advantage of residual connections with dense connections in the deep structure.

Method comparison

In this section, the proposed method is compared with the state-of-the-art methods. Here, Bicubic is viewed as the baseline. VSRNet (Kappeler et al. 2016), VESPCN (Caballero et al. 2017), SPMC (Tao et al. 2017), STCN (Guo and Chao 2017), Liu et al. (Liu et al. 2018), LapSRN (Lai et al. 2018), BRCN (Huang, Wang, and Wang 2018), and RDN (Zhang et al. 2018) are introduced. To verify the effectiveness of sparse fusion, RISTN without fusion (RISTN-NF) and RISTN with dense fusion (RISTN-DF) are introduced.

Figure 8: Temporal profiles for "Walk" from Vid4.

The experimental results are shown in Table 2. The proposed method outperforms all the other state-of-the-art approaches. Specifically, RISTN achieves highest evaluation

Figure 9: The visual comparisons on "Walk" of Vid4 with scaling factors as $4\times$.

scores and outperforms STCN, BRCN and Liu et al. by 1.22 dB / 0.058, 1.70 dB / 0.009 and 0.25 dB / 0.025 on PSNR/SSIM. The main reasons are that RISTN has a deeper architecture (RISTN is more than 130 convolutional layers, while STCN, BRCN only have 23 and 3 convolutional layers.) with both the spatial and temporal components, and it also fuses the spatial and temporal features for the final video frame reconstruction. The visual comparisons on "Calendar" and "Walk" are provided in Figure 7 and Figure 9. Besides, RISTN also gets better results compared with RISTN-NF, which demonstrates the effectiveness of the feature fusion. In comparison with RISTN-DF, the PSNR of RISTN rises by 0.15 dB, which demonstrates the effectiveness of the proposed sparse strategy. Follow by (Sajjadi, Vemulapalli, and Brown 2018), we adopt temporal profile to evaluate temporal consistency. RISTN gets finer details compared with other methods, as shown in Figure 8.

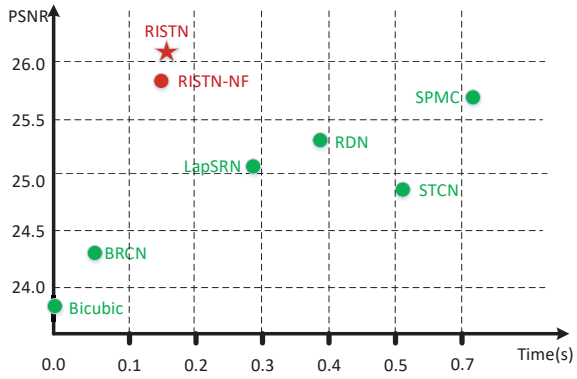


Figure 10: The average running time and the PSNR score.

The Figure 10 exhibits the efficiency comparison of average running time each frame. Obviously, the RISTN achieves promising performance of both the running time and PSNR. Although, the BRCN spends less time on processing each video frame, it leads to an unsatisfied PSNR result for the shallow architecture (3-layers). Compared with STCN and SPMC, RISTN shows superior performance. RISTN slightly slows the computing process of RISTN-NF for the computational costs of more channels in the fusion step. It should be discussed that FRVSR (Sajjadi, Vemulapalli, and Brown 2018) is current state-of-the-art VSR method uses both the optical flow and RGB features to capture motion and appearance information. But we only exploit appearance features to capture their relations.

Implementation details

The proposed network is trained specially for $4\times$ scale factor super-resolution. We randomly crop the 200×200 patch in each frame as the ground truth, and downsample it to 50×50 as the input LR patch for training. In the spatial component, the RGB video frames are zero pad to 16 channels. In the spatial component of RISTN, each parallel RIB branch contains four consecutive RIBs. In one of the RIB branch, there are 6, 8, 10, and 10 convolutional bottlenecks in each RIB; In the other RIB branch, there are 6, 6, 12 and 6. The last concatenated of the spatial component includes 256-channel feature maps in total. In the temporal component, the RDC-LSTM contains 10-bidirectional layers and the growth-rate is set as 16 (Huang et al. 2017). It outputs 320-channel feature maps in total. In the reconstruction component, the 320-channel temporal feature maps are converted to 256-channel feature maps. After fusion, the output has 256-channel feature maps in total. RISTN end-to-end is optimized by Adam with the learning rate 0.0001. The λ of L1 regular term is set as 5×10^{-7}. The training process is stopped when the training reaches 400 epochs and we select the best model for comparison. Experiments are performed on a NVIDIA Titan Xp GPU.

Conclusion

In this paper, we propose a novel residual invertible spatio-temporal network (RISTN) for effective and efficient video

super-resolutions. A lightweight residual invertible block (RIB) is proposed to reduce the information loss and it provides spatially consistent feature representations in the spatial component. The residual dense convolutional LSTM (RDC-LSTM) is designed for catching the temporal consistency and building a deep spatial feature transformation in the temporal component. The spatial and temporal feature maps are fused by a well-designed sparse strategy in the reconstruction component. Experimental results conducted on benchmark datasets show that RISTN achieves state-of-the-art performance.

Acknowledgement

This work was supported by National Key R&D Program of China (2017YFB1401000) and National Natural Science Foundation of China (61806044, 61602517, 61871378).

References

Caballero, J.; Ledig, C.; Aitken, A. P.; Acosta, A.; Totz, J.; Wang, Z.; and Shi, W. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*.

Demirel, H., and Anbarjafari, G. 2011. Discrete wavelet transform-based satellite image resolution enhancement. *IEEE Trans. Geoscience and Remote Sensing* 49(6-1):1997–2004.

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2016. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 38(2):295–307.

Fransens, R.; Strecha, C.; and Gool, L. V. 2004. A probabilistic approach to optical flow based super-resolution. In *CVPR Workshops*.

Gomez, A. N.; Ren, M.; Urtasun, R.; and Grosse, R. B. 2017. The reversible residual network: Backpropagation without storing activations. In *NIPS*.

Graves, A.; Jaitly, N.; and Mohamed, A. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *ASRU Workshop*.

Guo, J., and Chao, H. 2017. Building an end-to-end spatial-temporal convolutional network for video super-resolution. In *AAAI*.

Han, W.; Chang, S.; Liu, D.; Yu, M.; Witbrock, M.; and Huang, T. S. 2018. Image super-resolution via dual-state recurrent networks. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*.

Huang, Y.; Wang, W.; and Wang, L. 2018. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 40(4):1015–1028.

Jacobsen, J.-H.; Smeulders, A. W.; and Oyallon, E. 2018. i-revnet: Deep invertible networks. In *ICLR*.

Kappeler, A.; Yoo, S.; Dai, Q.; and Katsaggelos, A. K. 2016. Video super-resolution with convolutional neural networks. *IEEE Trans. Computational Imaging* 2(2):109–122.

Kim, J.; Lee, J. K.; and Lee, K. M. 2016a. Accurate image super-resolution using very deep convolutional networks. In *CVPR*.

Kim, J.; Lee, J. K.; and Lee, K. M. 2016b. Deeply-recursive convolutional network for image super-resolution. In *CVPR*.

Lai, W.; Huang, J.; Ahuja, N.; and Yang, M. 2018. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 1–1.

Liu, C., and Sun, D. 2011. A bayesian approach to adaptive video super resolution. In *CVPR*.

Liu, D.; Wang, Z.; Fan, Y.; Liu, X.; Wang, Z.; Chang, S.; Wang, X.; and Huang, T. S. 2018. Learning temporal dynamics for video super-resolution: A deep learning approach. *IEEE Trans. Image Processing* 27(7):3432–3445.

Ma, Z.; Liao, R.; Tao, X.; Xu, L.; Jia, J.; and Wu, E. 2015. Handling motion blur in multi-frame super-resolution. In *CVPR*.

Mitzel, D.; Pock, T.; Schoenemann, T.; and Cremers, D. 2009. Video super resolution using duality based TV-L1 optical flow. In *Pattern Recognition*, 432–441.

Rubinstein, R.; Zibulevsky, M.; and Elad, M. 2010. Double sparsity: learning sparse dictionaries for sparse signal approximation. *IEEE Trans. Signal Processing* 58(3):1553–1564.

Sajjadi, M. S. M.; Vemulapalli, R.; and Brown, M. 2018. Frame-recurrent video super-resolution. In *CVPR*.

Tai, Y.; Yang, J.; and Liu, X. 2017. Image super-resolution via deep recursive residual network. In *CVPR*.

Tao, X.; Gao, H.; Liao, R.; Wang, J.; and Jia, J. 2017. Detail-revealing deep video super-resolution. In *ICCV*.

Tong, T.; Li, G.; Liu, X.; and Gao, Q. 2017. Image super-resolution using dense skip connections. In *ICCV*.

Yang, W.; Feng, J.; Xie, G.; Liu, J.; Guo, Z.; and Yan, S. 2018. Video super-resolution based on spatial-temporal recurrent residual networks. *Computer Vision and Image Understanding* 168:79–92.

Zhang, L.; Zhang, H.; Shen, H.; and Li, P. 2010. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing* 90(3):848–859.

Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2018. Residual dense network for image super-resolution. In *CVPR*.