Spatiotemporal Super-Resolution with Cross-Task Consistency and its Semi-supervised Extension

Han-Yi Lin[1] , **Pi-Cheng Hsiu**[2,3] , **Tei-Wei Kuo**[1,4] and **Yen-Yu Lin**[2,5]
[1]Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taiwan
[3]Department of Computer Science and Information Engineering, National Chi Nan University, Taiwan
[4]Department of Computer Science and College of Engineering, City University of Hong Kong, Hong Kong
[5]Department of Computer Science, National Chiao Tung University, Taiwan
d03922006@csie.ntu.edu.tw, pchsiu@citi.sinica.edu.tw, ktw@csie.ntu.edu.tw, lin@cs.nctu.edu.tw

Abstract

Spatiotemporal super-resolution (SR) aims to upscale both the spatial and temporal dimensions of input videos, and produces videos with higher frame resolutions and rates. It involves two essential sub-tasks: spatial SR and temporal SR. We design a two-stream network for spatiotemporal SR in this work. One stream contains a temporal SR module followed by a spatial SR module, while the other stream has the same two modules in the reverse order. Based on the interchangeability of performing the two sub-tasks, the two network streams are supposed to produce consistent spatiotemporal SR results. Thus, we present a cross-stream consistency to enforce the similarity between the outputs of the two streams. In this way, the training of the two streams is correlated, which allows the two SR modules to share their supervisory signals and improve each other. In addition, the proposed cross-stream consistency does not consume labeled training data and can guide network training in an unsupervised manner. We leverage this property to carry out semi-supervised spatiotemporal SR. It turns out that our method makes the most of training data, and can derive an effective model with few high-resolution and high-frame-rate videos, achieving the state-of-the-art performance. The source code of this work is available at https://hankweb.github.io/STSRwithCrossTask/.

1 Introduction

Videos with high spatiotemporal resolutions are typically spatially sharper and temporally coherent, and hence are preferable to humans. However, acquiring such videos requires higher power consumption and larger storage. To compromise between the user experience and acquiring cost, spatial SR (super-resolution) and temporal SR have drawn increasing attention in computer vision. Spatiotemporal SR involves two sub-tasks, including *spatial SR* and *temporal SR*. Specifically, the former recovers the high-resolution frame by referring to a single low-resolution one, while the latter upscales the video frame rate by synthesizing intermediate frames.

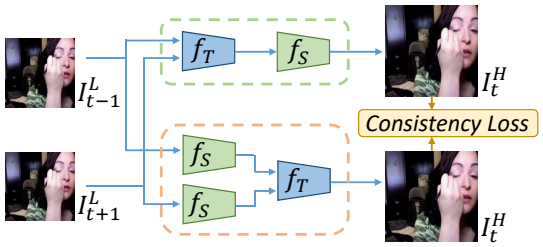


Figure 1: Spatiotemporal super-resolution (SR) aims to estimate the high-resolution frame I_t^H given its two temporally adjacent low-resolution frames, I_{t-1}^L and I_{t+1}^L. It involves two sub-tasks, spatial SR f_S and temporal SR f_T. Two network streams (green and brown blocks) are built by stacking modules f_S and f_T. The order of performing f_S and f_T is interchangeable. We leverage this property to develop a cross-stream consistency loss, which regularizes network training and more importantly enables semi-supervised learning.

In this work, we address spatiotemporal SR to increase the frame rate and resolution simultaneously for a given video. Both spatial SR such as [Shi *et al.*, 2016; Kim *et al.*, 2016; Haris *et al.*, 2018] and temporal SR such as [Liu *et al.*, 2017; Jiang *et al.*, 2018; Bao *et al.*, 2019] have been investigated extensively in the literature. However, few research advancements [Shahar *et al.*, 2011; Sharma *et al.*, 2017] have been made on spatiotemporal SR, which is more practical for low-quality video processing and understanding. An effective and intuitive way for spatiotemporal SR is to apply the two sub-tasks, spatial SR and temporal SR, sequentially. We consider the two sub-tasks highly correlated. On the one hand, spatial SR offers high-resolution frame details, which are essential to temporal SR. On the other hand, temporal SR enriches motion information, which generally facilitates spatial SR. However, the correlation between the two tasks has not been well exploited, especially in the era of deep learning.

This paper addresses the aforementioned issues. We correlate spatial SR and temporal SR for spatiotemporal SR based on a key insight: The order of applying spatial and temporal SR is interchangeable, as illustrated in Figure 1. We learn a two-stream network. One stream is composed of a temporal SR module f_T followed by a spatial SR module f_S for spatiotemporal SR. The other stream is formed by switching the two modules. No matter whether the ground truth for spatiotemporal SR is available, the results predicted by

the two streams should be consistent. Hence, we introduce a cross-stream consistency loss to enforce the similarity between the outputs of the two streams. The introduced loss correlates spatial SR and temporal SR. High-resolution details estimated by spatial SR help align pixels cross consecutive frames, which is crucial for temporal SR. High-frame-rate videos produced by temporal SR in turn facilitate spatial SR by regularizing it to predict temporally consistent SR results. Thereby, the two SR modules can improve each other by mutually providing supervisory signals.

More importantly, the cross-stream consistency loss does not rely on any high-resolution and high-frame-rate video for training. That is, learning from unlabeled training data, i.e., videos of low resolutions and frame rates here, is enabled. By taking abundant unlabeled videos as input, the cross-stream consistency loss serves as the objective function to guide network training in a self-taught manner. We leverage this property to carry out semi-supervised learning. It turns out that our method makes the most of training data, and can derive an effective model with few high-quality video collections.

The main contribution of this work is three-fold. First, we present the first end-to-end trainable network for spatiotemporal SR where the spatial and temporal SR modules are correlated and benefit each other. Second, the proposed method exploits cross-stream consistency and enables learning from unlabeled data, greatly reducing the cost of collecting videos of high spatiotemporal resolution, which can be very expensive in some applications like medical and satellite images. Third, evaluated intensively on four benchmark datasets including the Vimeo-90K [Xue *et al.*, 2019], Middlebury optical flow [Baker *et al.*, 2007], Vid4 [Liu and Sun, 2011], and DAVIS [Pont-Tuset *et al.*, 2017] datasets, our method achieves the state-of-the-art performance.

2 Related Work

2.1 Spatial Super-Resolution

Spatial SR has been explored extensively. Early approaches are developed based on the sampling theory, e.g., using linear or bicubic interpolation for SR [Keys, 1981]. However, interpolation exhibits limitations in predicting detailed, realistic textures. Advanced methods aim to establish complex mapping between low-resolution (LR) and high-resolution (HR) images by using machine learning algorithms, such as neighbor embedding [Gao *et al.*, 2012] and sparse coding [Zeyde *et al.*, 2012].

The pioneering work SRCNN [Dong *et al.*, 2016] employs a three-layer CNN model to approximate the non-linear mapping between LR and HR images in an end-to-end trainable manner. Based on residual learning [He *et al.*, 2016] for deep network optimization, modern CNN-based models such as VDSR [Kim *et al.*, 2016] and DRRN [Tai *et al.*, 2017] further boost the performance of SR. Recently, residual-dense network (RDN) [Zhang *et al.*, 2018] employs dense connections to learn the local representations from patches for improving SR. Likewise, DBPN [Haris *et al.*, 2018] presents a series of densely connected upsampling and downsampling layers to represent different image degradations for enhancing SR.

2.2 Temporal Super-Resolution

Conventional methods, e.g., [Baker *et al.*, 2011; Yu *et al.*, 2013], for temporal SR usually estimate dense correspondences between consecutive frames, and synthesize intermediate frames by the estimated correspondences. The quality of the interpolation results is highly dependent on the quality of the estimated optical flow. However, optical flow estimation is difficult and often suffers from many issues, such as occlusions, large motion, and blur.

CNNs have shown their effectiveness for optical flow estimation. Thereby, some CNN-based methods carry out frame interpolation by optical flow estimation [Bailer *et al.*, 2019]. However, these CNN-based methods relying on flow field prediction need training data in the form of dense correspondences, which are hard to annotate. Instead of relying on optical flow, some frame synthesis methods leverage CNNs to directly generate the intermediate frames [Niklaus *et al.*, 2017]. They do not take dense correspondences as training data but the ground-truth intermediate frames. However, these methods still suffer from blurred results and artifacts. Liu et al. [Liu *et al.*, 2017] address the problem of blurred results by proposing the deep voxel flow, a 3D optical flow to warp frames based on trilinear sampling. Their method makes the synthesized frames sharper, but the issue of artifacts remains unsolved. Recently, DAIN [Bao *et al.*, 2019] is proposed to explicitly detect occlusions by exploring depth cues, based on the observation that closer objects should be preferably synthesized in intermediate frames.

2.3 Spatiotemporal Super-Resolution

Research efforts on spatial SR and temporal SR are quite extensive. However, few advancements have been made on simultaneous spatial and temporal SR, i.e., spatiotemporal SR, which upscales video frame rates and resolutions at the same time. Spatiotemporal SR is important in many vision applications such as surveillance where analyzing videos of low quality is required. In the conventional approach, Shahar et al. [Shahar *et al.*, 2011] propose an example-based approach where spatiotemporal SR is realized by combining information from multiple space-time patches. Sharma et al. [Sharma *et al.*, 2017] propose the first deep-learning-based method, called *coupled deep convolutional auto-encoder* (CDCA), for spatiotemporal SR. CDCA generates the convolutional feature maps of the spatial patches in up-sampled LR and HR video frames using *convolutional auto-encoder* (CAE) and learns the relationships between these feature maps by CNNs at the same time. CDCA computes the up-sampled LR by tri-cubic interpolation, but it ignores motion correspondences between consecutive LR frames.

Unlike existing methods where the collaboration between the spatial and temporal modules is ignored, our method mitigates the aforementioned drawbacks by introducing the cross-stream consistency loss to jointly learn spatial and temporal SR modules in an end-to-end trainable manner. Thus, our method has the following advantages. First, the spatial SR and temporal SR modules are correlated and trained at the same time. More realistic frames are interpolated by referring to the high-resolution details recovered by the spatial SR

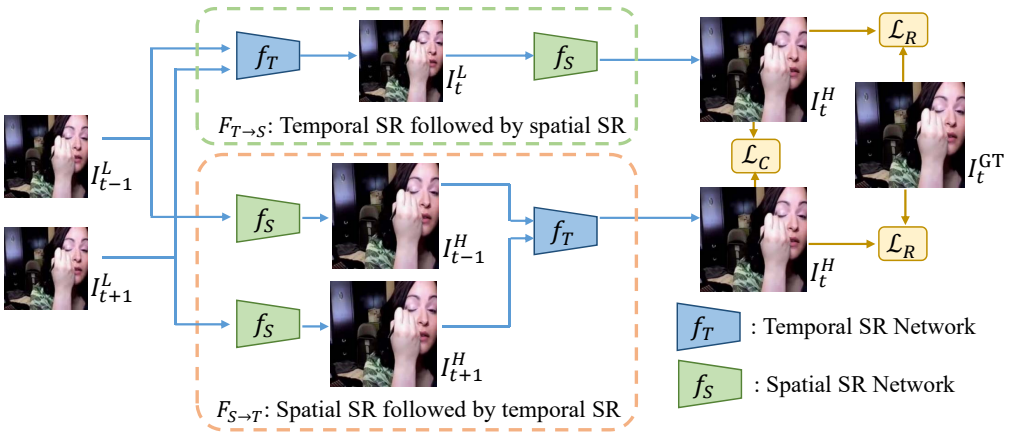


Figure 2: **Network architecture.** The proposed network architecture consists of two learnable modules, including f_S for spatial SR and f_T for temporal SR. It is a two-stream network with streams $F_{T\to S}$ and $F_{S\to T}$. Both streams contain the two learnable modules but have them in different orders. When the ground truth I_t^{GT} is given, the reconstruct loss $\mathcal{L}_R$ minimizes the difference between the ground truth and the prediction made by the two network streams. No matter if the ground truth I_t^{GT} is provided, the predictions made by the two streams are supposed to the same. Thus, the cross-stream consistency loss $\mathcal{L}_C$ is introduced to enforce the consistency between the outputs of the two streams. After training, each stream can perform spatiotemporal SR.

module, while better high-resolution frames are produced by using the motion information estimated by the temporal SR module. Second, the proposed cross-stream consistency loss allows learning from unlabeled data and greatly reduces the requirement of high-resolution videos. In addition, the spatial and temporal SR modules are present several times in the network, but the multiple copies of each module share weights, making the number of parameters manageable. These nice properties distinguish our method from prior work.

2.4 Cycle Consistency

Exploiting cycle consistency properties to regularize structured prediction has been explored in the literature. In unsupervised domain adaptation, exploiting cross-domain invariance in the label space achieves more consistent task predictions [Chen *et al.*, 2019]. In video frame interpolation, the cycle consistency loss enforces the similarity between the input frames and the cyclic mapped-back frames [Liu *et al.*, 2019]. In semantic matching, object co-segmentation, and co-saliency detection, cycle or transitivity based consistency losses help regularize the network training, e.g., [Chen *et al.*, 2015; Chen *et al.*, 2018; Chen *et al.*, 2020; Tsai *et al.*, 2019]. In motion analysis, computing bi-directional optical flow is useful to infer occlusions [Zou *et al.*, 2018] and enforce temporal consistency [Lai *et al.*, 2018].

In this work, we show a novel and feasible way of exploiting cross-stream consistency to address spatiotemporal SR. To the best of our knowledge, this work makes the first attempt to improve spatiotemporal SR by leveraging cross-stream consistency and end-to-end training. We design a two-stream network with spatial and temporal SR modules. Both streams consist of the two SR modules but have them in different orders. We show that enforcing the two streams to make consistent predictions leads to substantially improved performance. Employing the cross-stream consistency loss further enables semi-supervised learning.

3 Proposed Approach

This section describes the proposed approach. First, we give the problem definition. Then, the proposed network architecture and its objective function as well as the semi-supervised extension are described. Finally, the implementation details are provided.

3.1 Problem Definition

Given a video with a low frame resolution and/or low frame rate, the goal of spatiotemporal SR is to generate a high-quality video with a higher frame resolution and frame rate. Let I_{t-1}^L and I_{t+1}^L be two consecutive low-resolution (LR) frames at timestamps $t-1$ and $t+1$. The spatiotemporal SR model $\mathcal{F}$ in this work is derived to generate three high-resolution (HR) frames, I_{t-1}^H, I_t^H, and I_{t+1}^H. We use the superscript and subscript to denote the frame resolution and timestamp, respectively. The model $\mathcal{F}$ for upsacling both the frame resolution and frame rate can be represented as

$$(I_{t-1}^H, I_t^H, I_{t+1}^H) = \mathcal{F}(I_{t-1}^L, I_{t+1}^L). \tag{1}$$

Spatiotemporal SR can be decomposed into two sub-tasks: spatial SR and temporal SR, which are associated with learnable modules f_S and f_T, respectively.

Sub-task spatial SR targets at recovering the HR details and producing sharper video frames. It estimates an HR frame $I^H \in \mathbb{R}^{sP\times sQ}$ given an LR one $I^L \in \mathbb{R}^{P\times Q}$, where P and Q are the frame height and width respectively and $s>1$ is the upscaling factor. The module f_S can be expressed as

$$I^H = f_S(I^L). \tag{2}$$

By applying f_S to an frame k times, the frame resolution is increased by a factor of s^k.

The other sub-task temporal SR aims at upscaling video frame rates. Given two consecutive frames, I_{t-1} and I_{t+1}, of an arbitrary resolution, the temporal SR module f_T generates

the intermediate frame I_t between the two input frames. The module f_T can be formulated as

$$I_t = f_T(I_{t-1}, I_{t+1}). \tag{3}$$

Repeatedly applying module f_T to a video k times upscales the frame rate by a factor of 2^k.

To produce an HR video in both its image resolution and frame rate through Eq. (1), frames I^H_{t-1} and I^H_{t+1} can be obtained by applying the spatial SR module f_S to I^L_{t-1} and I^L_{t+1} respectively, while frame I^H_t relies on by the collaboration of both spatial SR module f_S and temporal SR module f_T. The two modules can work independently. One naive way for carrying out spatiotemporal SR is the sequential combination of the spatial and temporal SR modules, namely applying f_T followed by f_S or its inverse. In Figure 2, $F_{T\to S}$ and $F_{S\to T}$ represent two network streams, each of which combines the two SR modules. Given two consecutive LR frames I^L_{t-1} and I^L_{t+1}, either $F_{T\to S}$ or $F_{S\to T}$ can generate the HR frame I^H_t as defined below

$$\begin{aligned} I^H_t &\triangleq F_{T\to S}(I^L_{t-1}, I^L_{t+1}) = f_S(f_T(I^L_{t-1}, I^L_{t+1})) \text{ or} \\ &\triangleq F_{S\to T}(I^L_{t-1}, I^L_{t+1}) = f_T(f_S(I^L_{t-1}), f_S(I^L_{t+1})). \end{aligned} \tag{4}$$

3.2 Network Architecture

To better accomplish spatiotemporal SR and even carry it out in a semi-supervised fashion, we present an end-to-end trainable network which is composed of two network streams $F_{T\to S}$ and $F_{S\to T}$, as illustrated in Figure 2. By taking two consecutive LR frames I^L_{t-1} and I^L_{t-1} as input, stream $F_{T\to S}$ feeds them to the temporal SR module f_T to generate the interpolated frame I^L_t and passes I^L_t to the spatial SR module f_S to synthesize the HR frame I^H_t. In stream $F_{S\to T}$, I^L_{t-1} and I^L_{t-1} are fed into f_S to produce HR frames I^H_{t-1} and I^H_{t-1}, which then serve as the input to f_T to obtain the interpolated HR result I^H_t. The multiple copies of each module share weights so that the number of learnable parameters in the proposed network is manageable. We do not make any assumption about the spatial and temporal SR modules f_S and f_T. Our method is general to work with existing spatial and temporal SR algorithms by adopting them as the modules f_S and f_T, and accomplishes spatiotemporal SR. In the inference phase, either $F_{T\to S}$ or $F_{S\to T}$ can obtain the result. We just need to pass through one of the two streams.

3.3 Objective Function

The objective function $\mathcal{L}$ for training the proposed network consists of two loss terms, including the reconstruction loss $\mathcal{L}_R$ and cross-stream consistency loss $\mathcal{L}_C$. The former $\mathcal{L}_R$ guides the two network streams, $F_{S\to T}$ and $F_{T\to S}$, to perform spatiotemporal SR by making their prediction as close to the ground truth as possible. The latter $\mathcal{L}_C$ enforces the consistency between the results predicted by the two network streams. The training objective function $\mathcal{L}$ is defined by

$$\mathcal{L} = \mathcal{L}_R + \lambda \mathcal{L}_C, \tag{5}$$

where λ is the hyperparameter used to control the relative importance between $\mathcal{L}_R$ and $\mathcal{L}_C$, which are detailed below.

Reconstruction Loss $\mathcal{L}_R$

To guide the training of the two network streams, $F_{T\to S}$ and $F_{S\to T}$, this loss enforces their predicted results to be consistent with the ground truth. For each training triplet $\{I^H_{t-1}, I^H_t, I^H_{t+1}\}$, we first down-sample I^H_{t-1} and I^H_{t+1} to yield I^L_{t-1} and I^L_{t+1} by bicubic interpolation, respectively. With I^L_{t-1} and I^L_{t+1}, the reconstruction loss $\mathcal{L}_R$ is formulated as

$$\begin{aligned} \mathcal{L}_R(I^L_{t-1}, I^H_t, I^L_{t+1}) &= \|F_{S\to T}(I^L_{t-1}, I^L_{t+1}) - I^H_t\|^2 \\ &+ \|F_{T\to S}(I^L_{t-1}, I^L_{t+1}) - I^H_t\|^2. \end{aligned} \tag{6}$$

Cross-stream Consistency Loss $\mathcal{L}_C$

Our key insight for cross-stream consistency loss is that while both streams $F_{S\to T}$ and $F_{T\to S}$ can estimate the spatiotemporal SR results, their prediction should be the same no matter if the ground truth I^H_t is given. We thus propose a cross-stream consistency loss $\mathcal{L}_C$ that synchronizes the outputs of the two network streams. Specifically, this loss $\mathcal{L}_C$ is given below

$$\begin{aligned} \mathcal{L}_C(I^L_{t-1}, &I^L_{t+1}) = \\ &\|F_{S\to T}(I^L_{t-1}, I^L_{t+1}) - F_{T\to S}(I^L_{t-1}, I^L_{t+1})\|^2. \end{aligned} \tag{7}$$

3.4 Semi-supervised Learning

As shown in Eq. (7), the cross-stream consistency loss does not rely on any high-resolution ground truth. It allows the proposed method to work with unsupervised (i.e., low-resolution here) training data and hence enables semi-supervised learning. Specifically, the resulting loss function for semi-supervised spatiotemporal SR is designed as

$$\mathcal{L} = \mathcal{L}_R(D_L) + \lambda \mathcal{L}_C(D_L \cup D_U), \tag{8}$$

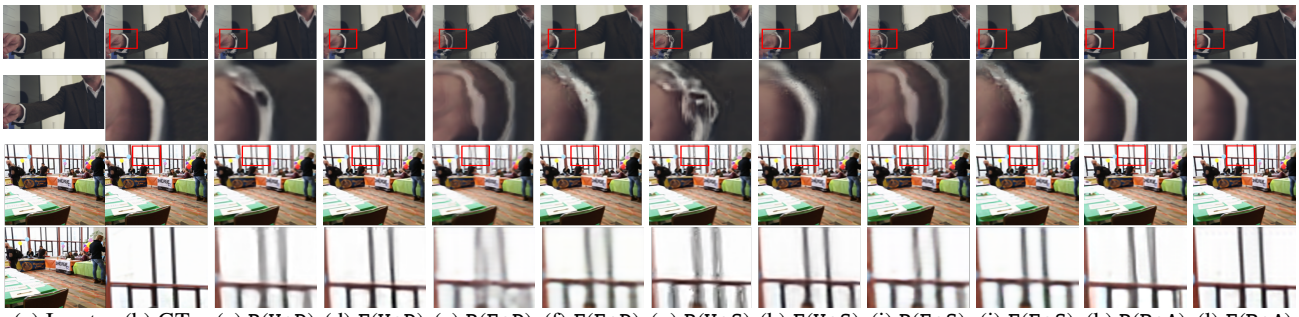
where the labeled data $D_L = \{(I^L_{t-1}, I^H_t, I^L_{t+1})\}$ are used in the reconstruction loss $\mathcal{L}_R$ and the cycle consistency loss $\mathcal{L}_C$. Unlabeled data $D_U = \{(I^L_{t-1}, I^L_{t+1})\}$ are used merely in $\mathcal{L}_C$ since D_U does not have high-resolution ground truth required in $\mathcal{L}_R$. By adopting the semi-supervised spatiotemporal SR, our model can be derived by using a small set of labeled data with a large amount of unlabeled data.

3.5 Implementation Details

Our method can work with existing spatial and temporal SR models. In this work, we use VDSR [Kim *et al.*, 2016], ESPCN [Shi *et al.*, 2016], and DBPN [Haris *et al.*, 2018] as the spatial SR module, while adopt DVF [Liu *et al.*, 2017], Super Slomo [Jiang *et al.*, 2018], and DAIN [Bao *et al.*, 2019] as the temporal SR module. Since DBPN and DAIN are released in Pytorch, our implementation regarding DBPN and DAIN is realized by Pytorch, while the rest are developed with Tensorflow. The spatial and temporal SR modules are first trained separately, then used to reconstruct the proposed network architecture, and finally fine-tuned. Although multiple spatial SR modules are present in the network, they are identical and share weights. This setting is also applied for the temporal SR module. We set the batch size, learning rate, momentum, and weight decay to 2, 10^{-3}, 0.9, and 5×10^{-4}, respectively. We train and evaluate our model on a single NVIDIA GeForce GTX 1080Ti graphics card with 11GB memory.

	Vimeo-90K						Vid4						Middlebury						DAVIS					
method	$F_{T\to S}$		$F_{S\to T}$		Avg.		$F_{T\to S}$		$F_{S\to T}$		Avg.		$F_{T\to S}$		$F_{S\to T}$		Avg.		$F_{T\to S}$		$F_{S\to T}$		Avg.	
metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
B(V+D)	28.91	0.847	28.84	0.849	28.87	0.848	22.20	0.661	22.03	0.654	22.11	0.657	25.42	0.712	25.19	0.705	25.30	0.709	22.33	0.665	22.18	0.660	22.25	0.663
F(V+D)	29.43	0.862	29.34	0.860	**29.39**	**0.861**	22.28	0.667	22.13	0.659	**22.21**	**0.663**	25.77	0.716	25.58	0.709	**25.68**	**0.713**	22.49	0.671	22.38	0.665	**22.43**	**0.668**
B(E+D)	27.37	0.789	28.51	0.830	27.94	0.810	22.20	0.645	22.68	0.686	22.44	0.665	27.02	0.751	27.13	0.764	27.07	0.757	23.02	0.697	22.63	0.691	22.82	0.694
F(E+D)	28.34	0.823	28.91	0.841	**28.63**	**0.832**	22.43	0.662	22.78	0.691	**22.60**	**0.676**	27.35	0.759	27.49	0.770	**27.42**	**0.765**	23.47	0.712	22.96	0.702	**23.22**	**0.707**
B(V+S)	28.46	0.835	28.45	0.836	28.45	0.835	22.63	0.682	22.63	0.685	22.63	0.684	26.55	0.754	26.49	0.755	26.52	0.755	22.67	0.692	22.58	0.691	22.63	0.691
F(V+S)	28.97	0.849	28.85	0.847	**28.91**	**0.848**	22.95	0.708	22.92	0.707	**22.94**	**0.708**	27.14	0.771	26.98	0.767	**27.06**	**0.769**	22.91	0.708	22.71	0.697	**22.81**	**0.702**
B(E+S)	28.17	0.826	27.12	0.797	27.64	0.811	22.08	0.642	22.44	0.674	22.26	0.658	26.56	0.746	25.32	0.721	25.94	0.733	22.60	0.690	21.66	0.653	22.13	0.671
F(E+S)	28.28	0.827	28.57	0.837	**28.42**	**0.832**	22.14	0.652	22.52	0.682	**22.33**	**0.667**	26.89	0.747	26.73	0.752	**26.81**	**0.750**	23.11	0.700	22.43	0.683	**22.77**	**0.692**
B(P+A)	28.79	0.855	29.23	0.875	29.01	0.865	22.03	0.671	22.43	0.695	22.23	0.683	26.67	0.747	27.16	0.775	26.91	0.761	23.50	0.727	23.61	0.736	23.55	0.731
F(P+A)	29.07	0.863	29.65	0.876	**29.36**	**0.870**	22.28	0.687	22.74	0.704	**22.51**	**0.695**	26.96	0.750	27.40	0.784	**27.18**	**0.767**	23.81	0.735	23.87	0.744	**23.84**	**0.740**

Table 1: Quantitative comparison between the baseline B and our method F with different spatial SR modules (including VDSR V, ESPCN E, and DBPN P) and temporal SR modules (DVF D, Super SloMo S, and DAIN A) on four test sets.



(a) Input (b) GT (c) B(V+D) (d) F(V+D) (e) B(E+D) (f) F(E+D) (g) B(V+S) (h) F(V+S) (i) B(E+S) (j) F(E+S) (k) B(P+A) (l) F(P+A)

Figure 3: Qualitative comparison between the baseline B and our method F with different spatial SR modules (VDSR V, ESPCN E, and DBPN P) and temporal SR modules (DVF D, Super SloMo S, and DAIN A) on two examples, each with input frames and ground truth.

4 Experimental Results

In this section, we first describe the datasets used in the experiments, and then conduct ablation studies and comparisons between our method and existing methods.

4.1 Datasets

We train the proposed method for spatiotemporal SR by using the training set of the Vimeo-90k dataset [Xue *et al.*, 2019], which is recently built for evaluating the performance of video processing tasks, such as video frame interpolation and super-resolution. The Vimeo-90k dataset contains $51,313$ samples for training and $3,782$ samples for testing. Each sample contains three high-resolution consecutive frames of resolution 256×448. For each training sample in the supervised setting, the middle frame serves as the ground truth while the low-resolution counterparts of the other two frames act as inputs. In our experiments, we downscale each frame side by a factor of 4 to yield the low-resolution counterparts. In the semi-supervised setting, only the low-resolution counterparts of unlabeled samples are used.

In addition to the testing set of Vimeo-90K, we use the Middlebury optical flow [Baker *et al.*, 2007], Vid4 [Liu and Sun, 2011], and DAVIS [Pont-Tuset *et al.*, 2017] datasets to evaluate the performance of the proposed method. For testing sets in the form of videos, we downscale the resolution of the odd-numbered frames by a factor of 4, and then remove all even-frames to generate low spatiotemporal resolution frame sequences. The downscaled consecutive odd-numbered frames serve as the inputs while the original even-numbered frames act as ground truth for evaluation. We adopt Peak Signal-to-Noise Ratio (PSNR) and structural similarity (SSIM) as the performance measure.

4.2 Ablation Studies

We conduct ablation studies for analyzing our method. Our method can work with existing spatial or temporal SR modules. We choose VDSR [Kim *et al.*, 2016], ESPCN [Shi *et al.*, 2016], and DBPN [Haris *et al.*, 2018] as the spatial SR modules, and select DVF [Liu *et al.*, 2017], Super Slomo [Jiang *et al.*, 2018], and DAIN [Bao *et al.*, 2019] as the temporal SR modules. Each baseline model is realized by sequentially applying the spatial and temporal models. Our method instead derives a two-stream model using the objective in Eq. (5). Both the baseline and our method are trained using the Vimeo-90K training set.

In the following, we compare our method to the baseline under different spatial and temporal module combinations. To measure the benefit of the proposed consistency loss $\mathcal{L}_C$, we also compare the results with and without using this loss. To analyze the proposed semi-supervised spatiotemporal SR, we perform the evaluation with various ratios of labeled data to the whole training set and different numbers of unlabeled data. Finally, we conduct the sensitivity analysis of hyper-parameter λ, and assess the effect of adopting cross-stream consistency loss $\mathcal{L}_C$.

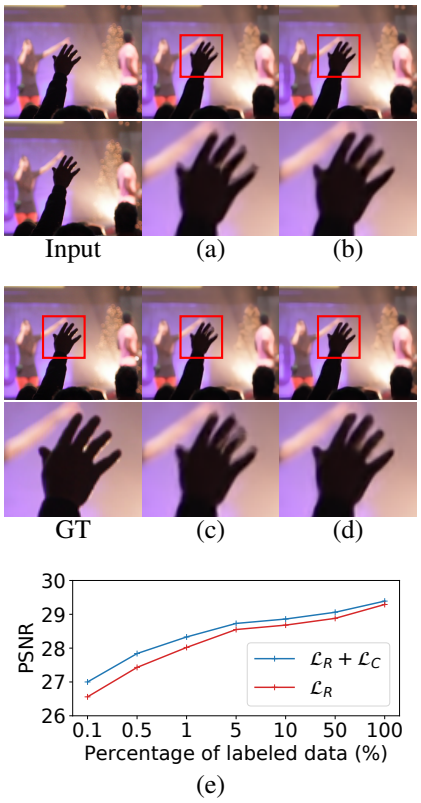


Figure 4: Frames synthesized by (a) $F_{S\to T}$, $\mathcal{L}_R$+$\mathcal{L}_C$; (b) $F_{T\to S}$, $\mathcal{L}_R$+$\mathcal{L}_C$; (c) $F_{S\to T}$, $\mathcal{L}_R$; (d) $F_{T\to S}$, $\mathcal{L}_R$.

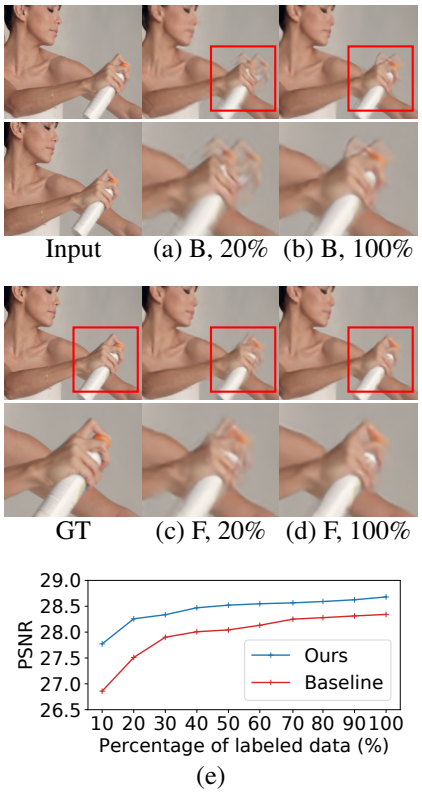


Figure 5: Frames synthesized by the baseline B and our method F using different amounts of labeled data.

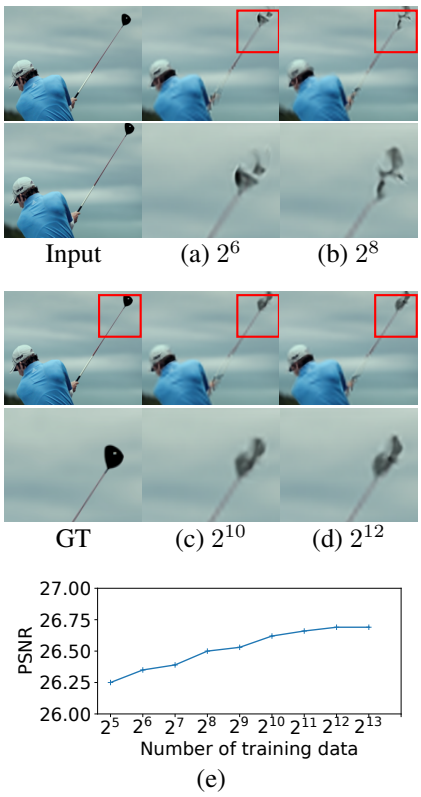


Figure 6: Synthesized frames and the performance with different numbers of unlabeled training data.

Comparisons with the baseline. We first verify whether the proposed network architecture has better performance by jointly learning the spatial and temporal SR modules across network streams. Table 1 reports the individual and the average qualities of spatiotemporal SR from the two network streams, $F_{S\to T}$ and $F_{T\to S}$, on four datasets. The results in Table 1 show that our proposed method achieves consistent performance gains over different module combinations and datasets. Figure 3 displays some spatiotemporal SR results synthesized by the baseline and the proposed method. Both the qualitative and quantitative results indicate that our method employing the two-stream network to correlate the spatial and temporal modules can produce remarkably better and sharper synthesized frames which contain less visual artifacts and exhibit characteristics more similar to the ground truth frames.

Cross-stream consistency loss $\mathcal{L}_C$. We verify if the cross-stream consistency loss $\mathcal{L}_C$ improves spatiotemporal SR under supervised and semi-supervised settings. Unless further specified, our method using VDSR and DVF as the spatial and temporal SR modules respectively is evaluated in the following experiments. Figure 4(e) shows the performance with and without $\mathcal{L}_C$ under different ratios of the labeled training data to the whole training set, from 0.1% to 100%. The results confirm that this loss $\mathcal{L}_C$ consistently improves the performance with different amounts of labeled training data.

Figure 4 visualizes the SR results with and without using $\mathcal{L}_C$ in the case of using 100% labeled training data. As shown in Figure 4(c) and 4(d), the absence of $\mathcal{L}_C$ leads to blurrier synthesized frames no matter which network stream, $F_{S\to T}$ or $F_{T\to S}$, is used for prediction. We also notice that $F_{S\to T}$ produces worse results. We consider that in this case of scenes with less texture and high motion, applying the spatial SR module first in stream $F_{S\to T}$ cannot recover useful HR details. Instead, firstly applying the temporal SR in $F_{T\to S}$ reveals this issue. However, which module should be applied first is unknown in general. As shown in Figure 4(a) and 4 (b), our method with the aid of $\mathcal{L}_C$ can alleviate the above problems to produce consistent results of two streams by enforcing the consistence between the two streams.

Two-stream network. We explore the impact of the ratio of labeled data to unlabeled data in semi-supervised learning. We randomly select 2,048 samples from Vimeo-90K as the training set, in which $k\%$ of these samples are labeled while the rest are unlabeled where $k \in \{10, 20, ..., 100\}$. Figure 5(e) shows that our method with two network streams achieves much better performance than the baseline with a single stream with various ratios. It is notable that our approach can use much less labeled data to reach the same performance by the baseline. Figure 5(b) and 5(c) show that our method with 20% of labeled data produces sharper and better frames than the baseline using 100% of labeled data.

(a) Input (b) Bicubic (c) Shahar et al. (d) CDCA (e) F(V+D) (f) F(E+D) (g) F(V+S) (h) F(E+S) (i) F(P+A)

Figure 7: Comparisons between the existing methods with our method F using different spatial and temporal modules including VDSR V, DVF D, DBPN P, ESPCN E, Super SloMo S, and DAIN A.

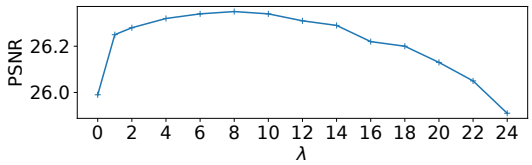


Figure 8: Sensitivity analysis of hyperparameter λ.

Input GT CDCA F(V+D) F(E+D) F(V+S) F(E+S) F(P+A)

Figure 9: Comparisons between CDCA and our method F.

	Vimeo-90K		Vid4		Middlebury		DAVIS	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CDCA	25.86	0.741	21.51	0.581	24.63	0.678	21.69	0.648
F(V+D)	29.39	0.861	22.21	0.663	25.68	0.713	22.43	0.668
F(E+D)	28.63	0.832	22.60	0.676	27.42	0.765	23.22	0.707
F(V+S)	28.91	0.848	22.94	0.708	27.06	0.769	22.81	0.702
F(E+S)	28.42	0.832	22.33	0.667	26.81	0.750	22.77	0.692
F(P+A)	29.36	0.870	22.51	0.695	27.18	0.767	23.84	0.740

Table 2: Comparisons between CDCA and our method F.

Effect of using unlabeled samples. To investigate the impact of unlabeled samples in semi-supervised spatiotemporal SR, we fix the number of labeled data to 32, and evaluate the performance of our method with different numbers of unlabeled data. Figure 6(e) shows that using the proposed cross-stream consistency loss to exploit unlabeled data improves the performance especially when more unlabeled data are provided. The visualization example in Figure 6 reveals that the synthesized frames become sharper and clearer when the number of unlabeled data increases.

Hyperparameter λ. We measure the effect of the hyperparameter λ, which controls the relative importance of the cross-stream consistency loss $\mathcal{L}_C$ to the reconstruction loss $\mathcal{L}_R$ in Eq. (5). We randomly select 512 samples from Vimeo-90K as the training set, in which 32 of these samples are labeled while the rest are unlabeled. Figure 8 reports the performance of our method with different values of hyperparameter λ. It can be observed that the loss function $\mathcal{L}_C$ is crucial, since the performance gain by changing λ from zero to a positive value is significant. Once the value of λ is larger than a threshold, say $8 \sim 10$ in this case, the performance decreases instead. The curve in Figure 8 is smooth. In addition, a broader range of the λ value results in the improved performance, which implies that finding a suitable value of λ to get performance gain is not difficult.

4.3 Comparisons with Existing Methods

The literature of spatiotemporal SR is limited. We compare our method with two existing methods: One is example-based method [Shahar *et al.*, 2011] and the other is a deep-learning-based method [Sharma *et al.*, 2017], called coupled deep convolutional auto-encoder (CDCA). For comparison with CDCA, we re-implement the approach. Note that the quantitative results of the method [Shahar *et al.*, 2011] are not available. We compare our method with that in [Shahar *et al.*, 2011] on the video data provided in its project website[1], and show qualitative comparison in the supplementary materials and Figure 7, where the proposed approach can produce more accurate and sharper frames, such as the regions of stars on the flag. Table 2 and Figure 9 show that our method performs favorably against CDCA on all the four datasets, and our method achieves the state-of-the art performance. CDCA ignores motion correspondences between consecutive low-resolution frames, leading to more artifacts especially in the regions with large motion. In contrast, our method leverages jointly the temporal SR module to smooth content transition and the spatial SR module to recover more high-resolution details, resulting in realistic and high-quality frames. About the inference time with the input two images of size 64×112 and the output image of size 256×448, our method takes about $60ms$, CDCA takes $18ms$. Although CDCA is more efficient, our method significantly outperforms CDCA.

[1] http://www.wisdom.weizmann.ac.il/~vision/SingleVideoSR.html

5 Conclusions

We present the first end-to-end trainable network for spatiotemporal SR. By exploiting cross-stream consistency to jointly train spatial and temporal SR modules, the proposed approach allows the two modules to share the supervisory signals and benefit each other. In addition, the cross-stream consistency loss enables semi-supervised learning, and can guide network training in a self-taught manner with unlabeled data. Both quantitative and qualitative results show that our method performs favorably against the existing methods, and achieving the state-of-the-art performance.

Acknowledgements

This work was supported in part by the Ministry of Science and Technology (MOST) under grants MOST 107-2628-E-009-007-MY3, MOST 109-2634-F-007-013, MOST 107-2628-E-001-001-MY3, and MOST 108-2218-E-002-048, and by Qualcomm through a Taiwan University Research Collaboration Project.

References

[Bailer *et al.*, 2019] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. *TPAMI*, 2019.

[Baker *et al.*, 2007] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007.

[Baker *et al.*, 2011] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011.

[Bao *et al.*, 2019] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, 2019.

[Chen *et al.*, 2015] Hsin-Yi Chen, Yen-Yu Lin, and Bing-Yu Chen. Co-segmentation guided hough transform for robust feature matching. *TPAMI*, 2015.

[Chen *et al.*, 2018] Yun-Chun Chen, Po-Hsiang Huang, Li-Yu Yu, Jia-Bin Huang, Ming-Hsuan Yang, and Yen-Yu Lin. Deep semantic matching with foreground detection and cycle-consistency. In *ACCV*, 2018.

[Chen *et al.*, 2019] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *CVPR*, 2019.

[Chen *et al.*, 2020] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation. *TPAMI*, 2020.

[Dong *et al.*, 2016] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2016.

[Gao *et al.*, 2012] Xinbo Gao, Kaibing Zhang, Dacheng Tao, and Xuelong Li. Image super-resolution with sparse neighbor embedding. *TIP*, 2012.

[Haris *et al.*, 2018] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Jiang *et al.*, 2018] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik G. Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018.

[Keys, 1981] Robert G Keys. Cubic convolution interpolation for digital image processing. *TASSP*, 1981.

[Kim *et al.*, 2016] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016.

[Lai *et al.*, 2018] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018.

[Liu and Sun, 2011] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *CVPR*, 2011.

[Liu *et al.*, 2017] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017.

[Liu *et al.*, 2019] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *AAAI*, 2019.

[Niklaus *et al.*, 2017] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017.

[Pont-Tuset *et al.*, 2017] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[Shahar *et al.*, 2011] Oded Shahar, Alon Faktor, and Michal Irani. Space-time super-resolution from a single video. In *CVPR*, 2011.

[Sharma *et al.*, 2017] Manoj Sharma, Santanu Chaudhury, and Brejesh Lall. Space-time super-resolution using deep learning based framework. In *PReMI*, 2017.

[Shi *et al.*, 2016] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

[Tai *et al.*, 2017] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, 2017.

[Tsai *et al.*, 2019] Chung-Chi Tsai, Weizhi Li, Kuang-Jui Hsu, Xiaoning Qian, and Yen-Yu Lin. Image co-saliency detection and co-segmentation via progressive joint optimization. *TIP*, 2019.

[Xue *et al.*, 2019] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019.

[Yu *et al.*, 2013] Zhefei Yu, Houqiang Li, Zhangyang Wang, Zeng Hu, and Chang Wen Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *TCSVT*, 2013.

[Zeyde *et al.*, 2012] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2012.

[Zhang *et al.*, 2018] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018.

[Zou *et al.*, 2018] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018.