

Robust Video Super-Resolution with Learned Temporal Dynamics

Ding Liu¹ Zhaowen Wang² Yuchen Fan¹ Xianming Liu³
 Zhangyang Wang⁴ Shiyu Chang⁵ Thomas Huang¹

¹University of Illinois at Urbana-Champaign ²Adobe Research

³Facebook ⁴Texas A&M University ⁵IBM Research

Abstract

Video super-resolution (SR) aims to generate a high-resolution (HR) frame from multiple low-resolution (LR) frames in a local temporal window. The inter-frame temporal relation is as crucial as the intra-frame spatial relation for tackling this problem. However, how to utilize temporal information efficiently and effectively remains challenging since **complex motion is difficult to model** and can introduce adverse effects if not handled properly. We address this problem from two aspects. First, we propose a **temporal adaptive neural network that can adaptively determine the optimal scale of temporal dependency**. Filters on various temporal scales are applied to the input LR sequence before their responses are adaptively aggregated. Second, we **reduce the complexity of motion** between neighboring frames using a **spatial alignment network** which is much more robust and efficient than competing alignment methods and can be jointly trained with the temporal adaptive network in **an end-to-end manner**. Our proposed models with learned temporal dynamics are systematically evaluated on public video datasets and achieve state-of-the-art SR results compared with other recent video SR approaches. Both of the temporal adaptation and the spatial alignment modules are demonstrated to considerably improve SR quality over their plain counterparts.

1. Introduction

Video super-resolution (SR) is the task of inferring a high-resolution (HR) video sequence from a low-resolution (LR) one. This problem has drawn growing attention in both the research community and industry recently. From the research perspective, this problem is challenging because video signals vary in both temporal and spatial dimensions. In the meantime, with the prevalence of high-definition (HD) display such as HDTV in the market, there is an increasing need for converting low quality video sequences to high-definition ones so that they can be played on the HD displays in a visually pleasant manner.

There are two types of relations that are utilized for video SR: the **intra-frame spatial relation** and the **inter-frame temporal relation**. In the past few years, neural network based models have successfully demonstrated the strong capability of modeling the spatial relation for image SR [4, 30, 5, 14, 15, 26, 6]. Compared with the intra-frame spatial relation, the inter-frame temporal relation is more important for video SR, as researches in vision system suggest that human vision system is more sensitive to motion [7]. Thus it is essential for video SR algorithm to capture and model the effect of motion information on visual perception. To meet this need, a number of video SR algorithms are proposed [8, 28, 2, 20, 23, 17] to conduct pixel-level motion and blur kernel estimation based on image priors, e.g., sparsity and total variation. These methods usually formulate a sophisticated optimization problem which requires heavy computational cost and thus is rather time-consuming to solve. Recently, neural network based models have also emerged in this domain [17, 9, 12]. Some of them model the temporal relation on a **fixed temporal scale** via explicitly conducting motion compensation to generate inputs to the network model [17, 12], while the rest develops **recurrent network architectures to use the long-term temporal dependency** [9].

As the key of temporal dependency modeling, motion estimation can influence video SR drastically. Rigid and smooth motion is usually easy to model among neighboring frames, in which case it is beneficial to include neighboring frames to super-resolve the center frame. In contrast, with **complex motion or parallax presented across neighboring frames**, motion estimation becomes challenging and erroneous motion compensation can undermine SR.

In this paper, we propose a **temporal adaptive neural network** that is able to **robustly handle various types of motion** and **adaptively select the optimal range of temporal dependency** to alleviate the detrimental effect of erroneous motion estimation between consecutive frames. Our network **takes as input a number of aligned LR frames after motion compensation**, and applies filters of different temporal sizes to generate multiple HR frame estimates. The

resultant HR estimates are adaptively aggregated according to the confidence of motion compensation which is inferred via another branch in our network. The proposed network architecture extends the idea of the Inception module in GoogLeNet [27] to the temporal domain. Our model gains the robustness to imperfect motion compensation **through network learning**, instead of simply boosting optical flow quality by using computationally more expensive methods as in [17], or extracting motion information only from a single fixed temporal scale as in [12].

Besides modeling motion information in the **temporal domain**, we can also compensate motion in the **spatial domain** to help the temporal modeling. We explore multiple image alignment methods to enhance video SR. We find the sophisticated optical flow based approach may not be optimal, as the estimation error on complex motion adversely affects the subsequent SR. Therefore, we **reduce the complexity of motion** by **estimating only a small number of parameters of spatial transform**, and provide a more robust approach to aligning frames. Moreover, inspired by the spatial transformer network [10], **we propose a spatial alignment network**, which **infers the spatial transform between consecutive frames** and **generates aligned frames for video SR**. It needs much less inference time and can be cascaded with the temporal adaptive network and trained jointly.

We conduct a systematic evaluation of each module in our network in the experiment section. The temporal adaptive design demonstrates a clear advantage in handling complex motion over the counterpart using temporal filters with fixed length. We observe that reducing the complexity of motion increases robustness of image alignment to complex motion and thus provides better SR performance. The spatial alignment network is proven to benefit SR by providing aligned input frames. Compared to other SR algorithms, our method not only yields state-of-the-art quantitative results on various video sequences, but also better recovers semantically faithful information benefiting high-level vision tasks.

2. Related Work

2.1. Deep Learning for Image SR

In the past few years, neural networks, especially convolutional neural networks (CNNs), have shown impressive performance for image SR. Dong et al. [4, 5] pioneer a three layer fully CNN, termed SRCNN, to approximate the complex non-linear mapping between the LR image and the HR counterpart. A neural network that closely mimics the sparse representation approach for image SR is designed by Wang et al. [30, 22], demonstrating the benefit of domain expertise from sparse coding in the task of image SR. A very deep CNN with residual architecture is proposed by Kim et al. [14] to attain impressive SR accuracy. Kim et al.

[15] design another network which has recursive architectures with skip-connection for image SR to boost performance while exploiting a small number of model parameters. Liu et al. [21] propose to learn a mixture of networks to further improve SR results. ESPCN proposed by Shi et al. [26] applies convolutions on the LR space of images and learns an array of upscaling filters in the last layer of their network model, which considerably reduces the computation cost and achieves real-time SR. More recently, Dong et al. [6] adopt a similar strategy to accelerate SRCNN with smaller filter sizes and more convolution layers.

2.2. Deep Learning for Video SR

With the popularity of neural networks for image SR, people have developed video SR methods of neural networks. Liao et al. [17] first generate an ensemble of SR draft via motion compensation under different parameter settings, and then use a CNN to reconstruct the HR frame from all drafts. Huang et al. [9] avoid explicit motion estimation by **extending SRCNN for single image SR along the temporal dimension forming a recurrent convolutional network to capture the long-term temporal dependency**. Kappeler et al. [12] expand SRCNN on a fixed temporal scale and extract features on frames aligned from optical flow information.

3. Temporal Adaptive Neural Network

3.1. Overview

For an LR video sequence, our model aims to estimate an HR frame from a set of local LR frames. The main challenge of video SR lies on the proper utilization of temporal information to handle various types of motion specifically. To address this problem, we design a neural network to adaptively select the optimal temporal scale for video SR. **The network has a number of SR inference branches $\{B_i\}_{i=1}^N$, each of which, B_i , works on a different temporal scale i , and uses its temporal dependency on its scale to predict an HR estimate.** We design an extra **temporal modulation branch**, T , to **determine the optimal temporal scale and adaptively combine all the HR estimates based on motion information**, at the pixel-level. All SR inference branches and the temporal modulation branch are incorporated and jointly learned in a unified network. **The final estimated HR frame is aggregated from the estimates from all SR inference branches considering the motion information on various temporal scales.** The overview of the temporal adaptive network is shown in Fig. 1.

3.2. Network Architecture

SR inference branch: We customize a recent neural network based SR model, ESPCN [26], due to its high SR accuracy and low computation cost, and use it in each SR in-

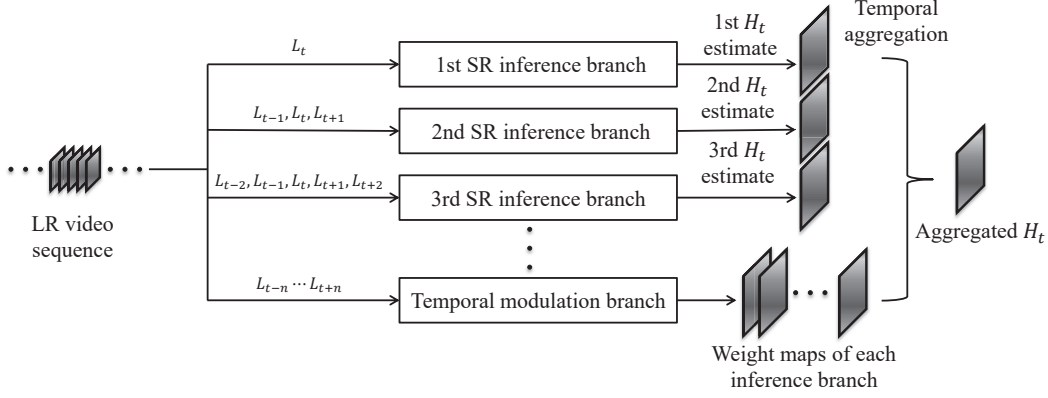


Figure 1. The overview of the temporal adaptive neural network. It consists of a number of SR inference branches and a temporal modulation branch. Each SR inference branch works on a different temporal scale and utilizes its temporal dependency to provide an HR frame prediction. These predictions are adaptively combined using pixel-level aggregations from the temporal modulation branch to generate the final HR frame.

inference branch. The SR inference branch B_i work on $2i - 1$ consecutive LR frames. c denotes the number of channels for every input LR frame. The filters of SR inference branch B_i in the first layer have the temporal length of $2i - 1$, and the convolutional filters in the first layer are customized to have $(2i - 1) \times c$ channels. The Rectified Linear Unit (ReLU) [24] is chosen as the activation function of the first and second layer. The interested readers are referred to [26] for the remaining details of ESPCN. Note that the design of the SR inference branch is not limited to ESPCN, and all other network based SR models, e.g. SRCNN, can work as the SR inference branch as well. The output of B_i serves as an estimate to the final HR frame.

Temporal modulation branch: The principle of this branch is to learn the selectivity of our model on different temporal scales according to motion information. We propose to assign pixel-level aggregation weights on each HR estimate, and in practice this branch is applied on the largest temporal scale. For a model of N SR inference branches, the temporal modulation branch takes $2N - 1$ consecutive frames as input. Considering the computation cost and efficiency, we adopt a similar architecture as the SR inference branch for this branch. The temporal modulation branch outputs the pixel-level weight maps on all N possible temporal scales.

Aggregation: Each SR inference branch's output is pixel-wisely multiplied with its corresponding weight map from the temporal modulation branch, and then these products are summed up to form the final estimated HR frame.

3.3. Training Objective

In training, we minimize the loss between the target HR frame and the predicted output, as

$$\min_{\Theta} \sum_j \|F(\mathbf{y}^{(j)}; \Theta) - \mathbf{x}^{(j)}\|_2^2, \quad (1)$$

where $F(\mathbf{y}; \Theta)$ represents the output of the temporal adaptive network, $\mathbf{x}^{(j)}$ is the j -th HR frame and $\mathbf{y}^{(j)}$ are all the associated LR frames; Θ is the set of parameters in the network.

If we use an extra function $W(\mathbf{y}; \theta_w)$ with parameter θ_w to represent the behavior of the temporal modulation branch, the cost function then can be expanded as:

$$\min_{\theta_w, \{\theta_{B_i}\}_{i=1}^N} \sum_j \left\| \sum_{i=1}^N W_i(\mathbf{y}^{(j)}; \theta_w) \odot F_{B_i}(\mathbf{y}^{(j)}; \theta_{B_i}) - \mathbf{x}^{(j)} \right\|_2^2. \quad (2)$$

Here \odot denotes the pointwise multiplication. $F_{B_i}(\mathbf{y}; \theta_{B_i})$ is the output of the SR inference branch B_i .

In practice, we first train each SR inference branch B_i individually as in (1) using the same HR frame as the training target, and then use the resultant models to initialize the SR inference branches when training the temporal adaptive network following (2). This training strategy speeds up the convergence dramatically without sacrificing the prediction accuracy of SR.

4. Spatial Alignment Methods

For video SR, people usually spatially align neighboring frames to increase the temporal coherence, and image alignment as a preprocessing step has proven beneficial to neural network based video SR methods [17, 12]. Therefore, we investigate several image alignment methods in order to provide better motion compensated frames for the temporal adaptive network.

4.1. Rectified Optical Flow Alignment

It is well known that since the complex motion is difficult to model, the conventional optical flow based image alignment using erroneous motion estimation may introduce artifacts, which can be propagated to the following SR step and have a detrimental effect on it. We try simplifying the

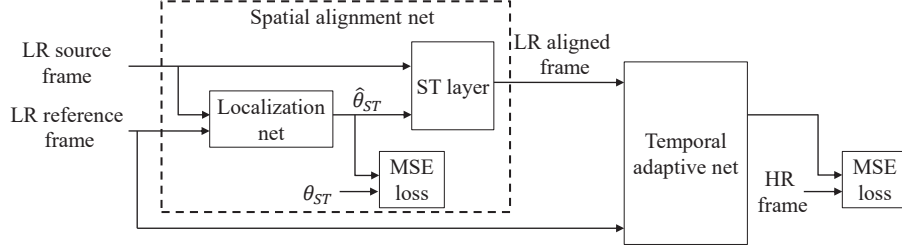


Figure 2. The architecture of the spatial alignment network and the cascade of it with the SR network. Each time **one LR reference frame** and **its neighboring LR frame** (as the source frame) are first fed into a localization net to regress the alignment parameter $\hat{\theta}_{ST}$. Then $\hat{\theta}_{ST}$ is **applied to the source frame in the spatial transform layer**, to generate an LR aligned frame. The **LR reference frame** and **all the aligned neighboring frames** are the input to the subsequent SR network. During the joint training, we **minimize the weighted sum of the MSE loss** in the SR network and **the MSE loss between $\hat{\theta}_{ST}$ and the ground truth θ_{ST}** in the spatial alignment network.

motion in the patch level to integer translations for avoiding interpolation which may cause blur or aliasing. Given a patch and its optical flow, we estimate the integer translation along the horizontal and vertical directions by **rounding the average horizontal and vertical displacement of all pixels in this patch**, respectively. This scheme, termed **rectified optical flow alignment**, proves more beneficial to the following SR than the conventional optical flow based image alignment, which will be shown in Section 5.4.

4.2. Spatial Alignment Network

In order to align neighboring frames, we propose a spatial alignment network which has the merits of efficient inference and end-to-end training with the SR network. The architecture of the spatial alignment network and the cascade of it with the temporal adaptive network are shown in Figure 2.

Each time the spatial alignment network takes as input one LR reference frame and one LR neighboring frame (as the source frame), and generates as output an aligned version of this neighboring frame. Specifically, first these two LR frames are fed into a localization network to predict the spatial transform parameter $\hat{\theta}_{ST}$, which is then applied to the source frame in the spatial transform layer, to produce the LR aligned frame. Given the success of the rectified optical flow alignment, we design the localization network to infer only two translation parameters. The LR source frame and reference frame are stacked to form the input of $2 \times c$ channels. The localization network has two convolutional layers of 32 kernels in the size of 9×9 . Each convolutional layer is followed by a max-pooling layer with stride of 2 and kernel size of 2. Then there are two fully connected layers which have 100 and 2 nodes, respectively, to regress the 2 translation parameters. In practice, the spatial alignment network works on the patch level for better motion modeling, and only the center region in each patch is kept in order to avoid the empty area near the boundary after translation. In training, the loss in the spatial alignment network is defined as the mean squared error (MSE) between

$\hat{\theta}_{ST}$ and the ground truth θ_{ST} , which can be acquired from other image alignment methods, e.g. the rectified optical flow alignment.

The LR reference frame and all the resultant aligned neighboring frames from this network are used together as input to the temporal adaptive network for SR.

We propose to train this network and the temporal adaptive network in an end-to-end fashion, since joint learning is usually more advantageous than separate learning. During the joint training, we minimize the weighted sum of the loss in the spatial alignment network and the loss from the temporal adaptive network, as

$$\min_{\{\Theta, \theta_L\}} \sum_j \|F(\mathbf{y}^{(j)}; \Theta) - \mathbf{x}^{(j)}\|_2^2 + \lambda \sum_j \sum_{k \in \mathcal{N}_j} \|\hat{\theta}_{ST}^{(k)} - \theta_{ST}^{(k)}\|_2^2, \quad (3)$$

where \mathcal{N}_j denotes the set of LR frames associated to the j -th HR frame, and λ is the scaling factor for balancing these two losses. θ_L represents the set of parameters in the localization net.

5. Experiments

5.1. Datasets

Since the amount of training data plays an important factor in training neural network, we combine three public datasets of uncompressed videos: LIVE Video Quality Assessment Database [25], MCL-V Database [18] and TUM 1080p Data Set [13], so as to collect sufficient training data. We prepare data in 3D volume from video clips that have abundant textures and details and are separated by shots or scenes. We choose six videos: *calendar*, *city*, *foliage*, *penguin*, *temple* and *walk* as in [17], and a 4k video dataset, Ultra Video Group Database [1], as the test sets. The original HR frames are downsized by bicubic interpolation to generate LR frames for training.

5.2. Implementation Details

Following the convention in [4, 30, 9, 26], we convert each frame into the YCbCr color space and only process

Table 1. PSNR (in dB) comparisons of different network architectures: average PSNR of all the frames in each video sequence by 4x upscaling is displayed. Best results are shown in bold. From left to right, B_i : the HR prediction from the i -th SR inference branch; $B_{1,2}$: the straight average of HR predictions from B_1 and B_2 ; $B_{1,2} + T$: the adaptive aggregation of the outputs of B_1 and B_2 with joint learning. $B_{1,2,3}$ and $B_{1,2,3} + T$ follow the similar definitions as $B_{1,2}$ and $B_{1,2} + T$, respectively.

	B_1	B_2	B_3	$B_{1,2}$	$B_{1,2} + T$	$B_{1,2,3}$	$B_{1,2,3} + T$
calendar	20.88	21.16	21.32	21.10	21.26	21.24	21.51
city	25.70	25.91	26.25	25.89	25.97	26.10	26.46
foliage	24.29	24.51	24.81	24.47	24.58	24.70	24.98
penguin	36.46	36.41	36.40	36.56	36.53	36.59	36.65
temple	28.97	29.30	29.72	29.33	29.46	29.64	30.02
walk	27.69	27.78	27.90	27.88	27.90	28.00	28.16
average	27.33	27.51	27.73	27.54	27.62	27.71	27.96

the luminance channel with our model. Hence each frame has $c = 1$ channel. We focus on the upscaling factor of four, which is usually the most challenging case in video SR. The input LR frames to the temporal adaptive network are the volume of $5 \times 30 \times 30$ pixels, i.e. patches of 30×30 pixels from five consecutive frames. These data are augmented with rotation, reflection and scaling, providing about 10 million training samples. We implement our model using Caffe [11]. We apply a constant learning rate of 10^{-4} for the first two layers and 10^{-5} for the last layer, a batch size of 64 with momentum of 0.9. We terminate the training after five million iterations. Experiments are conducted on a workstation with one GTX Titan X GPU. Based on the architecture of each SR inference branch, we can initialize the parameters from the single frame SR model, except that the filter weights in the first layer are evenly divided along the temporal dimension. In practice, it is observed that this initialization strategy leads to faster convergence and usually improves the performances.

5.3. Analysis of Network Architecture

We investigate the SR performance of different architectures of the temporal adaptive network. Recall that B_i denotes the SR inference branch working on $2i - 1$ LR frames, and T is the temporal adaptive branch. We explore the architectures that contain (1) only B_1 , (2) only B_2 , (3) only B_3 , (4) $B_{1,2}$, (5) $B_{1,2} + T$, (6) $B_{1,2,3}$ and (6) $B_{1,2,3} + T$. $B_{1,2}$ denotes the straight average of HR predictions from B_1 and B_2 , and $B_{1,2,3}$ follows the similar definition. Note that in the case of (1), each frame is super-resolved independently. For the experiments in this section, LR consecutive frames are aligned as in Section 4.1.

The PSNR (unit: dB) comparisons of six test sequences by 4x upscaling are shown in Table 1. Average PSNR of all the frames in each video is shown in the table. It can be observed that generally the network performance is enhanced as more frames are involved, and $B_{1,2,3} + T$ performs the best among all the architectures. $B_{1,2} + T$ obtains higher PSNRs than $B_{1,2}$ and $B_{1,2,3} + T$ is superior than $B_{1,2,3}$, which demonstrates the advantage of adaptive aggregation over the straight averaging on various temporal scales.



Figure 3. Examples of SR results from *walk* by 4x upscaling using different network architectures. Compared with B_1 and B_3 , the temporal adaptive architecture $B_{1,2,3} + T$ is able to effectively handle both the rigid motion shown in the top left zoom-in region and the complex motion in the bottom right zoom-in region.

In order to show the visual difference of SR results among various architectures, we choose one frame from *walk*, and show the SR results from B_1 , B_3 , $B_{1,2,3} + T$ as well as the ground truth HR frame in Figure 3 with two zoom-in regions. The region in the blue bounding box contains part of a flying pigeon which is subject to complex motion among consecutive frames and thus is challenging for accurate motion estimation. It can be observed that the HR inference from B_1 has much less artifacts than that from B_3 , indicating the short term temporal dependency alleviates the detrimental effect of erroneous motion estimation in this case. On the contrary, the zoom-in region in the red bounding box includes the ear and part of the neck of the pedestrian with nearly rigid motion, in which case the HR inference from B_3 is able to recover more details. This manifests the necessity of the long term temporal dependency. $B_{1,2,3} + T$ is able to generate better HR estimates compared with its counterparts of single branch in these two zoom-in regions, and thus shows the effectiveness of the temporal adaptive design in our model.

To further analyze the temporal adaptive branch, we vi-

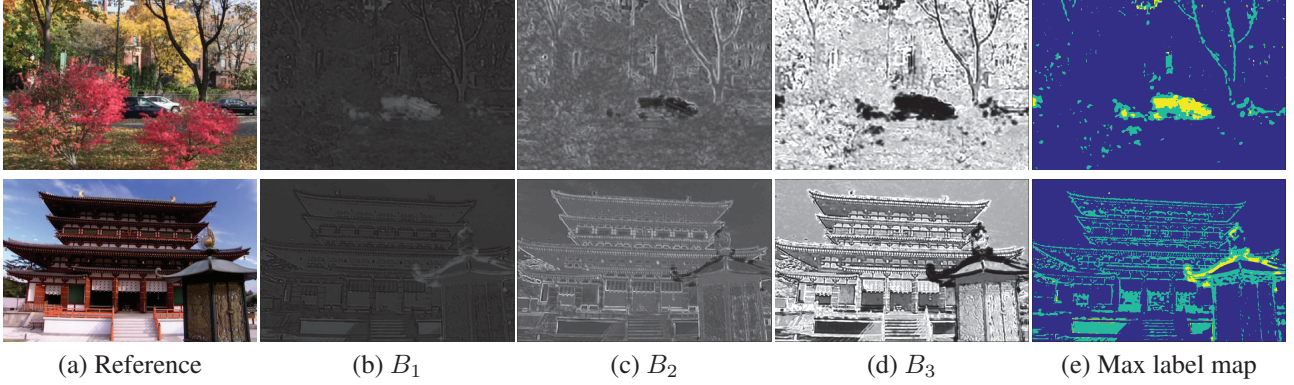


Figure 4. Weight maps of three SR inference branches given by the temporal modulation branch in the architecture $B_{1,2,3} + T$. The max label map records the index of the maximum weight among all the SR inference branches at every pixel, which is shown in the last column. B_1 , B_2 and B_3 are indicated in yellow, teal and blue, respectively. Frames from top to bottom: *foliage* and *temple* by 4x upscaling.

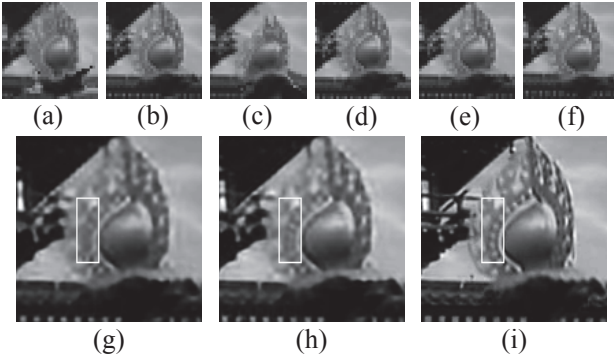


Figure 5. Comparison between the optical flow alignment and our proposed rectified optical flow alignment. (a)-(c): three consecutive LR patches warped by optical flow alignment. (d)-(f): the same three LR patches aligned by rectified optical flow alignment. (g): HR prediction from (a)-(c). (h): HR prediction from (d)-(f). (i): ground truth. Note that the rectified optical flow alignment recovers more details inside the white bounding box.

sualize the weight maps given by the temporal modulation branch of two frames from *foliage* and *temple* for each SR inference branch. In addition, we use the index of the maximum weight among all SR inference branches at each pixel to draw a max label map. These results are displayed in Figure 4. It can be seen that B_1 mainly contributes the region of cars in *foliage* and the top region of the lantern in *temple*, which are subject to large displacements caused by object motion and camera motion. On the contrary, the weight map of B_3 has larger responses in the region subject to rigid and smooth motion, such as the plants of the background in *foliage* and the sky in *temple*. Their complementary behaviors are properly utilized by the temporal adaptive aggregation of our model.

5.4. Analysis of Spatial Alignment Methods

We conduct experiments with the image alignment methods discussed in Section 4. We choose the algorithm of Liu [19] to calculate optical flow, considering the motion estimation accuracy and running time. Figure 5 shows a case

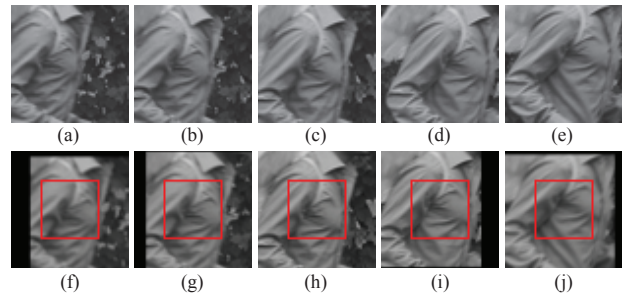


Figure 6. Visualization of the inputs and outputs of the spatial alignment network. (a)-(e): five consecutive LR frames as inputs. (f)-(j): their corresponding outputs. Note that (c) and (h) are the same reference frame without processing of this network. Only the region in the center red bounding box is kept for SR.

where the conventional optical flow alignment fails and introduces obvious artifacts in LR frames while the rectified optical flow alignment succeeds.

As for the spatial alignment network, we use the result of the rectified optical flow alignment as the training target of the spatial transform parameter. We choose the input patch size as 60×60 pixels for the spatial alignment network to achieve the best motion modeling. We keep the center region to be 30×30 pixels in each patch and discard the rest region after translation. In the joint training λ is set as -10^3 empirically. The visualization of the inputs and outputs of the spatial alignment network is shown in Figure 6. It is obvious that the output neighboring frames are aligned to the reference frame.

The PSNR comparisons between these alignment methods on six video sequences by 4x upscaling are shown in Table 3. Average PSNR of all the frames in each video is shown, and only B_3 is used in the SR network. Our proposed rectified optical flow alignment achieves the highest PSNR, demonstrating its superiority over the conventional optical flow alignment. The approach of spatial alignment network clearly improves SR quality over its plain counterpart, which shows the effectiveness of its alignment.

Table 2. PSNR (in dB) comparisons of various video SR methods: PSNR of only the center frame in each video sequence by 4x upscaling is displayed. Best results are shown in bold.

	VSRnet[12]	Bayesian[20]	Deep-DE[17]	ESPCN[26]	VDSR[14]	Proposed
calendar	20.99	21.59	21.40	20.97	21.50	21.61
city	24.78	26.23	25.72	25.60	25.16	26.29
foliage	23.87	24.43	24.92	24.24	24.41	24.99
penguin	35.93	32.65	30.69	36.50	36.60	36.68
temple	28.34	29.18	29.50	29.17	29.81	30.65
walk	27.02	26.39	26.67	27.74	27.97	28.06
average	26.82	26.75	26.48	27.29	27.58	28.05

Table 3. PSNR (in dB) comparisons of various frame alignment methods: average PSNR of all the frames in each video sequence by 4x upscaling is displayed. Best results are shown in bold. From left to right, raw: raw LR frames; OF: conventional optical flow alignment; ROF: rectified optical flow alignment; SAN: spatial alignment network.

	raw	OF	ROF	SAN
calendar	21.07	21.28	21.32	21.27
city	25.92	26.29	26.25	26.23
foliage	24.49	24.62	24.81	24.78
penguin	36.37	36.32	36.40	36.29
temple	29.40	29.52	29.72	29.60
walk	27.82	27.79	27.90	27.83
average	27.51	27.64	27.73	27.67

5.5. Comparison with State-of-the-Art

We use our best model $B_{1,2,3} + T$ with rectified optical flow alignment to compare with several recent image and video SR methods: VSRnet [12], Bayesian method [20], Deep-DE [17], ESPCN [26] and VDSR [14] on the six test sequences. We use the model and code of VSRnet, Deep-DE and VDSR from their websites, respectively. The source code of Bayesian method is unavailable, so we adopt the reimplement of Bayesian method in [23] and use five consecutive frames to predict the center frame. We implement ESPCN by ourselves since its source code is unavailable as well. We report the result on only the center frame of each sequence in that Deep-DE requires 15 preceding and 15 succeeding frames to predict one center frame and there are only 31 frames in each sequence. We display several visual results in Figure 7. It can be seen that our method is able to recover more fine details with shaper edges and less artifacts. The PSNRs are shown in Table 2. Our method achieves the highest PSNR on all the frames.

For the application of HD video SR, we compare our method with VSRnet and ESPCN on Ultra Video Group Database. We do not include Bayesian method and Deep-DE for comparison, since both of them take multiple hours to predict one 4k HR frame (only the CPU version of code for Deep-DE is available). The PSNR comparisons of these three methods are in Table 4. Average PSNR of all the frames in every video is shown. Our method obtains the

Table 4. PSNR (in dB) comparisons of several video SR methods on Ultra Video Group Database by 4x upscaling. Best results are shown in bold.

	VSRnet[12]	ESPCN[26]	Proposed
Beauty	35.46	35.68	35.72
Bosphorus	43.02	43.01	43.28
HoneyBee	39.69	39.82	39.89
Jockey	40.25	40.65	40.81
ReadySteadyGo	39.69	40.36	40.82
ShakeNDry	39.06	39.51	39.58
YachtRide	37.48	37.56	37.87
Average	39.24	39.52	39.71

Table 5. Face identification accuracy of using various video SR methods on YouTube Face dataset downsampled by a factor of 4. The baseline refers to the result from the model directly trained on LR frames. Best results are shown in bold.

	Top-1 accuracy	Top-5 accuracy
Baseline	0.442	0.709
VSRnet[12]	0.485	0.733
ESPCN[26]	0.493	0.734
Proposed	0.511	0.762

highest PSNR consistently over all the video sequences.

SR can be used as a pre-processing step to enhance the performance of high-level vision applications, such as semantic segmentation, face recognition and digit recognition [3, 29], especially when the input data is of low visual quality. Here we evaluate how various video SR algorithms could benefit the video face identification on YouTube Face (YTF) dataset [31]. Our proposed method is compared with VSRnet and ESPCN. We form a YTF subset by choosing the 167 subject classes that contain more than three video sequences. For each class, we randomly select one video for testing and the rest for training. The face regions are cropped and resized to 60×60 pixels, as the original resolution set, and then are downsampled by a factor of 4 to comprise the low-resolution set. We train a customized AlexNet [16] on the original resolution set as the classifier, and feed SR results from the low-resolution set by various algorithms for face identification. During testing, the prediction probability is aggregated over all the frames in each video clip. The top-1 and top-5 accuracy results of face identification are reported in Table 5. We include as the

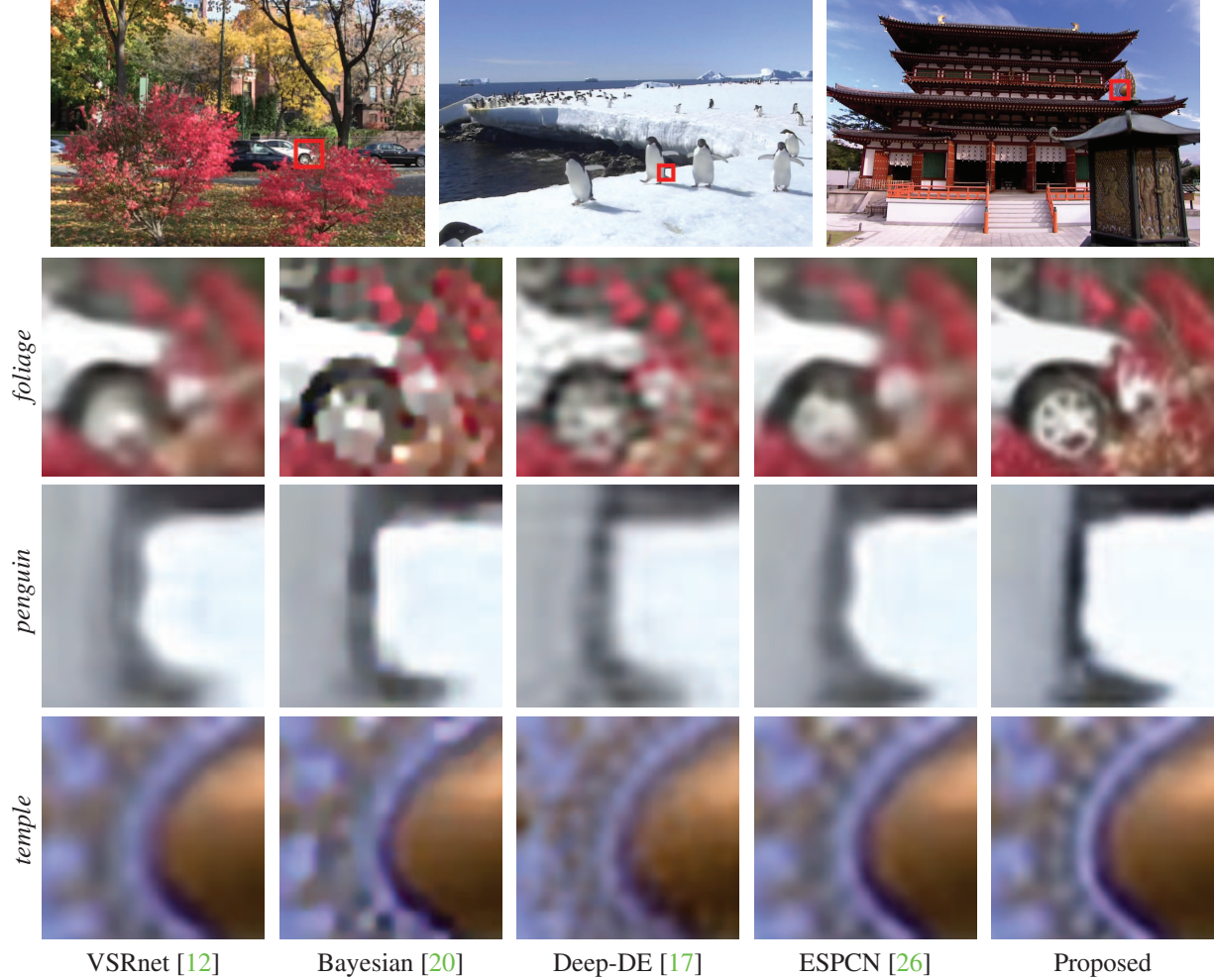


Figure 7. Visual comparisons of SR results by 4x upscaling among different methods. From top to bottom: zoom-in regions from *foliage*, *penguin* and *temple*.

baseline the result from a model directly trained on the LR set. Our proposed method achieves both the highest top-1 and top-5 accuracy among all the SR methods, showing that it is able to not only produce visually pleasing results, but also recover more semantically faithful features that benefit high-level vision tasks.

5.6. Running Time Analysis

In testing, the running time is mainly composed of two parts: the frame alignment as pre-processing and the SR inference. For the first part, the spatial alignment network can align frames significantly faster than the optical flow based method. For 4x SR of 4k videos, it takes about 15s to warp five consecutive frames of 540×960 pixels for the optical flow based method on an Intel i7 CPU, while the spatial alignment network needs only around 0.8s on the same CPU, which reduces the time by one order of magnitude. For the second part, B_1 takes about 0.3s to generate a 4k HR frame. B_1 , B_2 and B_3 differ only in the numbers of channels in the first layer, so their inference time varies a

little. The inference time of the temporal modulation branch is comparable to that of the SR inference branch. All these branches enjoy the benefit of extracting features directly on LR frames and can be implemented in parallel for time efficiency.

6. Conclusions

In this paper, we propose a temporal adaptive network and explore several methods of image alignment including a spatial alignment network, for the better usage of temporal dependency and spatial alignment to enhance video SR. We compare our proposed model with other recent video SR approaches comprehensively on various video sequences and our model obtains state-of-the-art SR performance. Both the temporal adaptation and the enhanced spatial alignment increase the robustness to complex motion which benefits video SR.

References

- [1] <http://ultravideo.cs.tut.fi/>. 4
- [2] S. P. Belekos, N. P. Galatsanos, and A. K. Katsaggelos. Maximum a posteriori video super-resolution using a new multi-channel image prior. *TIP*, 19(6):1451–1464, 2010. 1
- [3] D. Dai, Y. Wang, Y. Chen, and L. Van Gool. Is image super-resolution helpful for other vision tasks? In *WACV*, pages 1–9. IEEE, 2016. 7
- [4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199. Springer, 2014. 1, 2, 4
- [5] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2016. 1, 2
- [6] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pages 391–407. Springer, 2016. 1, 2
- [7] M. Dorr, T. Martinetz, K. R. Gegenfurtner, and E. Barth. Variability of eye movements when viewing dynamic natural scenes. *Journal of vision*, 10(10):28–28, 2010. 1
- [8] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *TIP*, 13(10):1327–1344, 2004. 1
- [9] Y. Huang, W. Wang, and L. Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *NIPS*, pages 235–243, 2015. 1, 2, 4
- [10] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 2
- [11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 5
- [12] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, 2016. 1, 2, 3, 7, 8
- [13] C. Keimel, J. Habigt, T. Habigt, M. Rothbucher, and K. Diepold. Visual quality of current coding technologies at high definition iptv bitrates. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 390–393. IEEE, 2010. 4
- [14] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1, 2, 7
- [15] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 1, 2
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 7
- [17] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia. Video super-resolution via deep draft-ensemble learning. In *CVPR*, pages 531–539, 2015. 1, 2, 3, 4, 7, 8
- [18] J. Y. Lin, R. Song, C.-H. Wu, T. Liu, H. Wang, and C.-C. J. Kuo. Mcl-v: A streaming video quality assessment database. *Journal of Visual Communication and Image Representation*, 30:1–9, 2015. 4
- [19] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Citeseer, 2009. 6
- [20] C. Liu and D. Sun. On bayesian adaptive video super resolution. *TPAMI*, 36(2):346–360, 2014. 1, 7, 8
- [21] D. Liu, Z. Wang, N. Nasrabadi, and T. Huang. Learning a mixture of deep networks for single image super-resolution. In *ACCV*, pages 145–156. Springer, 2016. 2
- [22] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang. Robust single image super-resolution via deep networks with sparse prior. *TIP*, 25(7):3194–3207, 2016. 2
- [23] Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, and E. Wu. Handling motion blur in multi-frame super-resolution. In *CVPR*, pages 5224–5232, 2015. 1, 7
- [24] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010. 3
- [25] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. Study of subjective and objective quality assessment of video. *TIP*, 19(6):1427–1441, 2010. 4
- [26] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, pages 1874–1883, 2016. 1, 2, 3, 4, 7, 8
- [27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 2
- [28] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super-resolution without explicit subpixel motion estimation. *TIP*, 18(9):1958–1975, 2009. 1
- [29] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang. Studying very low resolution recognition using deep networks. In *CVPR*, pages 4792–4800, 2016. 7
- [30] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *ICCV*, pages 370–378, 2015. 1, 2, 4
- [31] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, pages 529–534. IEEE, 2011. 7