




# Space-Time Super-Resolution Using Deep Learning Based Framework

Manoj Sharma<sup>(✉)</sup> , Santanu Chaudhury , and Brejesh Lall 

Department of Electrical Engineering, Indian Institute of Technology,  
Delhi 110016, India

mksnith@gmail.com, schaudhury@gmail.com, brejesh@ee.iitd.ac.in

**Abstract.** This paper introduces a novel end-to-end deep learning framework to learn space-time super-resolution (SR) process. We propose a coupled deep convolutional auto-encoder (CDCA) which learns the non-linear mapping between convolutional features of up-sampled low-resolution (LR) video sequence patches and convolutional features of high-resolution (HR) video sequence patches. The upsampling in LR video refers to tri-cubic interpolation both in space and time. We also propose a H.264/AVC compatible video space-time SR framework by using learned CDCA, which enables to super-resolve compressed LR video with less computational complexity. The experimental results prove that the proposed H.264/AVC compatible framework performs better than the state-of-art techniques on space-time SR in terms of quality and time complexity.

**Keywords:** Deep learning · Image and video super-resolution · Space-time super-resolution · H.264/AVC

## 1 Introduction

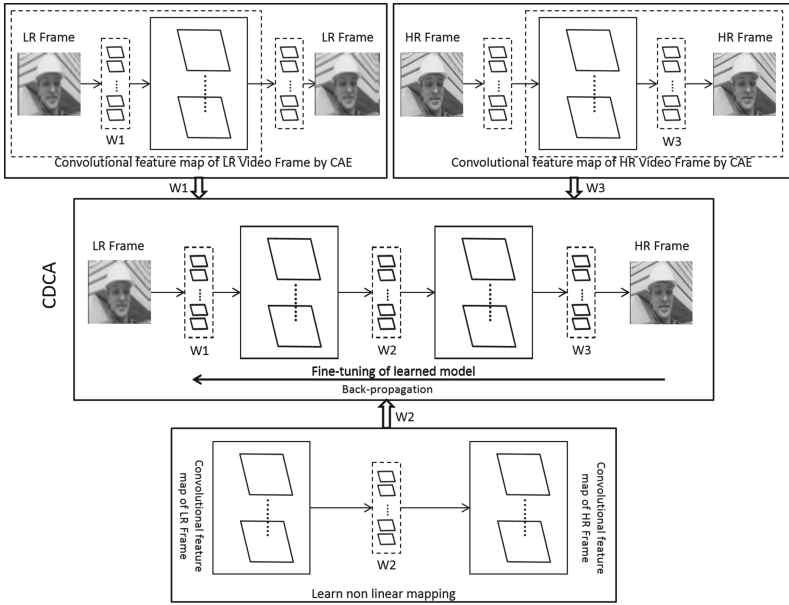
Super-resolution (SR) of videos can be categorized into spatial SR and temporal SR. The recovery of HR video frames from LR video frames is termed as spatial SR. On the other hand, temporal SR is the retrieval of those dynamic events which occur faster than provided frame-rate by predicting mid-frame information. Although much work has been done on natural images SR [1–4] and spatial SR [5–8] of videos, but few advancement have been made to achieve simultaneous space-time SR. This problem is more interesting and useful in many computer vision and biomedical tasks for pre-processing of videos. One class of space-time SR methods [9, 10] takes multiple LR video sequences at the input. Another class of space-time SR methods is to super-resolve video in space and time using only single video [11]. Existed work on space-time SR from single LR video doesn't provide significant improvement. There is still much scope for simultaneous space-time resolution enhancement.

In this paper, we propose a novel deep learning based method which we call *coupled deep convolutional auto-encoder (CDCA)* to learn the relationship

between up-sampled (both in space and time by tri-cubic interpolation) LR and corresponding HR video sequence spatial patches. The proposed method simultaneously calculates the convolutional feature map of up-sampled LR and HR video frame spatial patches using convolutional auto-encoder (CAE) and learns the relationship between these feature maps using the convolutional neural network. Our framework is motivated by a machine learning-based method [2] for natural images SR, we adopt similar framework with some major improvement for space-time SR. In contrast to autoencoder used by Zeng et al. [2] that computed intrinsic features, convolutional auto-encoder (CAE) was used in our framework to extract the features since CAE provides a better representation of image patches [12] in comparison to simple auto-encoder. Additionally, in our framework, the convolutional neural network is used to learn the mapping between convolutional features of LR and HR patches. We learn the mapping between spatial patches of up-sampled LR and HR video sequences instead of 3D space-time patches for the optimization of computation complexity. Learning on 3D space-time patches will provide better high temporal frequency information at the cost of high computational complexity. We also extend the use of CDCA to propose H.264/AVC compatible framework, which enables to super-resolve videos in a compressed domain with less computing complexity. Existing works on video super-resolution were limited to raw videos, but almost all of the videos on the web and other sources are encoded (compressed) due to bandwidth and memory limitation. First, one has to convert encoded video into the raw video, then that video can be super-resolved. But, our proposed architecture can directly super-resolve encoded video during the decoding (decompression) process with less computational complexity, which makes it more suitable for real-time space-time video SR.

## 2 Space-Time Super-Resolution Using CDCA

Our LR video sequence has a dimension ( $W \times L \times T$ ) and corresponding HR video sequence and up-sampled LR have a dimension ( $S.W \times S.L \times S.T$ ). Here  $S$  is the space-time SR factor. The CDCA given in Fig. 1 has a three-stage architecture. In Fig. 1, we term up-sampled LR frame as LR frame. The first and third stage consist of two convolutional auto-encoder (CAE) to learn the convolutional feature map of up-sampled LR (both in space and time) and corresponding HR video frame spatial patches, respectively. This results in weights/filters of CAE to learn useful features which can reconstruct back the original video frame spatial patches. After that, we make the algorithm to learn the non-linearities between LR and HR video frame spatial patches convolutional feature map by using simple one layer convolutional neural network (CNN) in the second stage. Here, weights/filters are learned to obtain the map between convolutional feature map of LR and HR video frame spatial patches. After having learned weights/filters of all stages, we put all three stages together to form one network as shown in Fig. 1. Then this network is fine-tuned on space-time super-resolution dataset which has up-sampled LR video frame spatial patches as input and HR video frame spatial patches as the target.



**Fig. 1.** Block Diagram of CDCA

We consider the up-sampled LR video frame spatial patches  $Y_i$  and learn mapping with corresponding HR video frame patches  $X_i : \forall i = 1, 2..n$  where  $n$  is the total number of patches in training database. As a pre-processing step, we normalize the patch elements between  $[0, 1]$ . Then convolutional feature map for LR video frame spatial patches is given by,

$$f_{i,L}^k = \max(0, W_1^k * Y_i + b_1^k) \tag{1}$$

and, LR video frame spatial patches are reconstructed back by convolutional feature map as,

$$\hat{Y}_i = \max(0, \sum_{k \in N} f_{i,L}^k * W_1'^k + c_1) \tag{2}$$

by, minimizing the loss function,

$$loss_{LR} = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \| Y_i - \hat{Y}_i \|^2 \tag{3}$$

Here  $N$  is the total number of feature maps of LR video frame spatial patches. Similarly, convolutional feature maps for HR video frame spatial patches is given by,

$$f_{i,H}^k = \max(0, W_3'^k * X_i + b_3^k) \tag{4}$$

reconstruction of HR video frame patches as,

$$\hat{X}_i = \max(0, \sum_{k \in M} f_{i,H}^k * W_3^k + c_3) \quad (5)$$

by, minimizing the loss function,

$$loss_{HR} = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \| X_i - \hat{X}_i \|_2^2 \quad (6)$$

here,  $M$  is the total number of convolutional feature maps of HR video frame spatial patches.  $M$  should be lesser than  $N$  ( $M \ll N$ ) to enforce sparsity and the relation between  $f_L^k$  and  $f_H^k$  is represented as,

$$\hat{f}_{i,H}^k = \max(0, W_2^k * f_{i,L}^k + b_2^k) \quad (7)$$

Mapping between  $f_L^k$  and  $f_H^k$  is learned by minimizing the loss function,

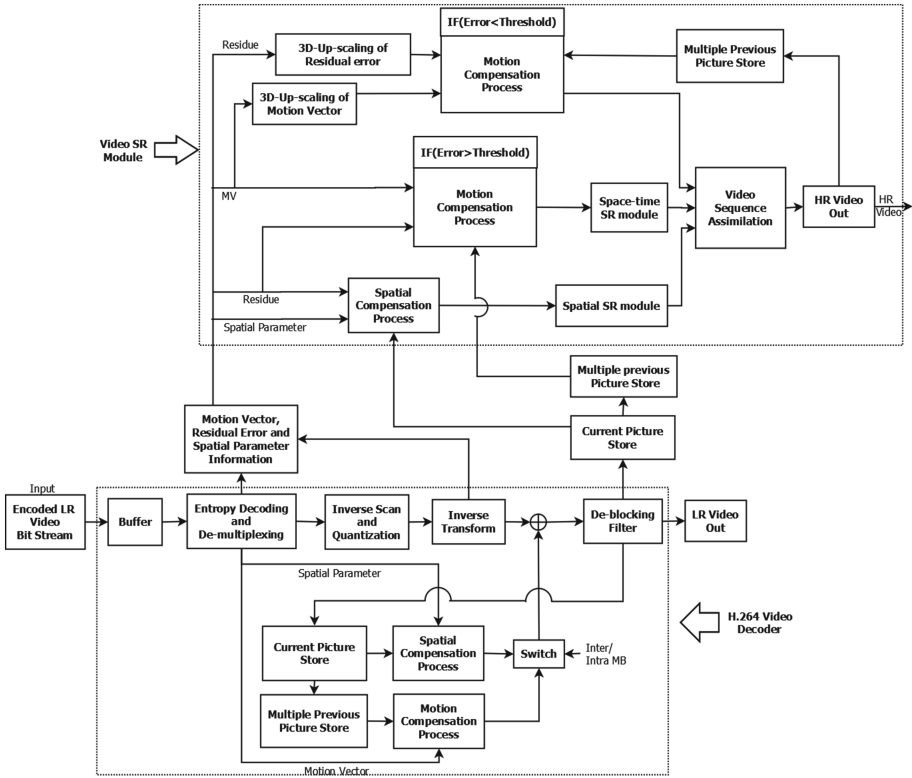
$$loss = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \| f_{i,H}^k - \hat{f}_{i,H}^k \|_2^2 \quad (8)$$

After having pre-trained CDCA parameters  $W_1, W_2, W_3, b_1, b_2, c_3$ , we fine-tune all the parameters of combined framework CDCA on space-time SR data-set.

## 2.1 H.264/AVC Compatible Framework for Space-Time SR

We propose a novel H.264/AVC Compatible video SR framework which is using space-time SR algorithm, motion vector, spatial prediction parameters and residual error information to get HR video sequence from compressed LR video bit-stream. The proposed framework is given in Fig. 2. Our proposed video space-time SR framework uses different approaches for super-resolving I slices and P/B slices macro-blocks for optimization of computational cost. The working of this framework is described below:

1. In Fig. 2, encoded LR video bit-stream is the input for the standard H.264 decoder.
2. The motion vector (MV), residual error, spatial parameter and previously stored frame information are extracted from standard H.264 video decoder and are given to video space-time SR module.
3. All macro-blocks of I-slices and intra-predicted macro-blocks of P and B-slices are spatially super-resolved by applying spatial SR module block after spatial compensation process as given in lower part of video SR module. These macro-blocks are super-resolved by adding residues to the spatially compensated macro-blocks and then, followed by spatial SR module (i.e. CDCA learned on SRCNN [1] training dataset).



**Fig. 2.** Block Diagram of H.264 Compatible Space-Time SR Framework

4. The P and B-slices inter-predicted macro-blocks are super-resolved in space and time by using information of residue, MV, and space-time SR module as given in Fig. 2.
5. If the residue is greater than the threshold (as shown by middle part of video SR module), then the HR macro-blocks sequences are inter-predicted by adding residues to motion compensated macro-blocks sequences, and then followed by space-time SR module (input to space-time SR module is 3D up-sampled LR macro-blocks sequences).
6. If the residue is less than a threshold (as given in upper part of video SR module), MV and residue are up-scaled (both in space and time). HR macro-blocks sequences are predicted by adding re-scaled residues to the motion compensated macro-blocks sequences.
7. All HR macro-blocks sequences are arranged together to form a super-resolved video sequence.
8. Super-resolved I frame are stored as a reference frame, to be used as a reference for future P and B-frames.

The framework is also compatible with HEVC encoding scheme since HEVC has almost similar compression and decompression framework as H.264 with some extra features like adaptive Loop Filter, more number of intra-predicted modes and DCT based interpolation for luminance etc.

### 3 Results

Video sequences are taken from [11] to generate training data-set. LR video sequences are generated by 3D (space-time) down-sampling of HR video sequences. For the comparison of the qualitative performance of our H.264/AVC compatible framework with the existing state-of-the-art video spatial super-resolution methods, we take test sequences similar to those used in [7] to compute the results. To verify the effectiveness of proposed framework for spatial video SR, we conducted experiments on different standard video sequences. We compared proposed framework with state-of-the-art video SR algorithms as shown in

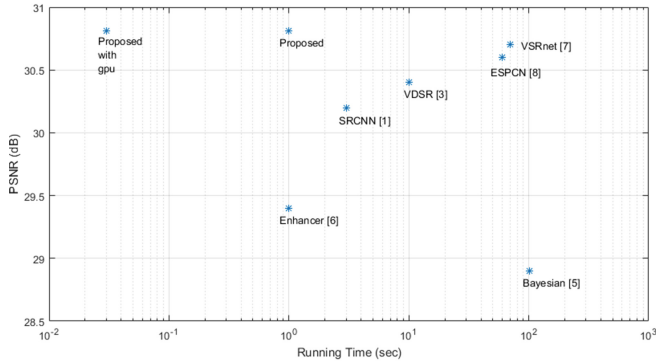


Fig. 3. Comparison between performance and runtime of different algorithms

Table 1. Average PSNR and SSIM comparison of different Video SR algorithms for different sequences

Sequence	Scale	Bayesian [5]		Enhancer [6]		VSRnet [7]		ESPCN [8]		Proposed	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Myanmar	2	35.56	0.9515	35.94	0.9588	<b>38.48</b>	<b>0.9679</b>	38.37	0.9605	38.42	0.9612
Myanmar	3	32.20	0.9203	32.50	0.9099	34.42	0.9247	34.31	0.9239	<b>34.50</b>	<b>0.9288</b>
Myanmar	4	30.68	0.8895	30.23	0.8681	31.85	0.8834	31.53	0.8816	<b>31.98</b>	<b>0.8871</b>
Videose4	2	29.69	0.9055	30.40	0.9141	31.30	0.9278	31.14	0.9237	<b>31.34</b>	<b>0.9300</b>
Videose4	3	25.82	0.8328	26.34	0.7948	<b>26.79</b>	<b>0.8098</b>	26.44	0.8023	26.74	0.8097
Videose4	4	25.06	0.7466	24.55	0.6877	24.84	0.7049	24.79	0.7008	<b>25.09</b>	<b>0.7178</b>
Foreman	2	35.88	0.9652	37.22	0.9693	38.52	0.9738	38.29	0.9721	<b>38.59</b>	<b>0.9758</b>
Foreman	3	33.81	0.9098	34.12	0.9105	35.74	0.9243	35.62	0.9234	<b>35.82</b>	<b>0.9334</b>
Foreman	4	33.26	0.8787	33.76	0.8808	34.69	0.8926	34.46	0.8879	<b>34.76</b>	<b>0.9095</b>

Table 1. Results show that proposed framework is comparable with state-of-the-art techniques. The experiment was conducted with a Linux work-station containing an Intel Xeon E5-2687W v3 processor with 3.1 GHz and 64 GB RAM. The graphics card used was NVIDIA GeForce GTX 980 with 2084 cores. In Fig. 3, we plot the average PSNR and run-time of different SR algorithms to 3× super-resolve per frame from the Myanmar and Videaset4 test sequence with a 704 × 576 resolution. Figure 3 clearly shows reduced computing complexity of our proposed framework in comparison of existing state-of-the-art video space-time SR techniques.

Motion aliasing occurs when the camera frame rate is lower than the temporal frequency of a fast moving object in video sequences. Observed object seems to be in false trajectory or distorted. Wagon wheel effect given in [11] is one of the best examples of motion aliasing effect; here the fan seems to be rotating in



Fig. 4. Temporal SR comparison (3×) between Bayesian approach (left) and our approach (right) [upper and lower one are key frames and mid frames are predicted one]



Fig. 5. Temporal SR (3×) on Flag sequence using (a) Tri-cubic interpolation. (b) [11]. (c) Proposed.

counter clock-wise direction [look at upper and lower key frames] but the actual rotation is in a clock-wise direction. This effect can be reduced by predicting mid-frame information. In Fig. 4, we show the comparison of our space-temporal SR approach with a Bayesian approach to reduce motion aliasing effect. We can easily visualize that quality of mid-frame is better and accurate in our approach. Some-times fast moving objects results in bad object shapes and blurriness along their motion trajectory. This effect becomes more prominent with an increase in object motion speed. It can be reduced by improving space-temporal SR. In Fig. 5, we show the comparison of different temporal SR approaches. Our framework is providing more visually pleasing video frames in comparison of exiting work and helps in reducing fast motion effects like, blurring and shape distortion.

## 4 Conclusions

We have proposed CDCA to learn space-time SR process and H.264/AVC compatible video space-time SR framework. Proposed H.264/AVC compatible framework outperforms all existing approaches for video spatial SR and space-time SR. The proposed framework drastically reduces the implementation complexity of space-time super-resolution learning algorithm in videos. This reduction in complexity and its implementation in GPU results in real-time space-time up-scaling of videos with improved quality.

## References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). doi:[10.1007/978-3-319-10593-2\\_13](https://doi.org/10.1007/978-3-319-10593-2_13)
2. Zeng, K., Yu, J., Wang, R., Li, C., Tao, D.: Coupled deep auto-encoder for single image super-resolution. *IEEE Trans. Cybern.* **47**(1), 27–37 (2015)
3. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
4. Mao, X.-J., Shen, C., Yang, Y.-B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: NIPS (2016)
5. Liu, C., Sun, D.: On bayesian adaptive video super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(2), 346–360 (2014)
6. Faroudja, Y.C.: Video Enhancer, Patent App. PCT/US1993/001,907 (1993)
7. Kappeler, A., Yoo, S., Dai, Q., Katsaggelos, A.K.: Video super-resolution with convolutional neural networks. *IEEE Trans. Comput. Imaging* **2**(2), 109–122 (2016)
8. Shi, W., Cabalero, J.: Real time single image and video super-resolution using an efficient sub-pixel convolutional neural networks. In: CVPR (2016)
9. Shechtman, E., Caspi, Y., Irani, M.: Space-time super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(4), 531–545 (2005)
10. Mudénagudi, U., Banerjee, S., Kalra, P.K.: Space-time super-resolution using graph cut optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 995–1008 (2011)



11. Themelis, K.E., Rontogiannis, A.A., Koutroumbas, K.D.: Space-time super-resolution from a single video. In: CVPR (2011)
12. Du, B., Xiong, W., Wu, J., Zhang, L., Tao, D.: Stacked convolutional de-noising auto-encoders for feature representation. *IEEE Trans. Cybern.* **47**(4), 1017–1027 (2016)