Deep Recurrent Resnet for Video Super-Resolution

Bee Lim and Kyoung Mu Lee
Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul, South Korea
Email: forestrainee@gmail.com, kyoungmu@snu.ac.kr

***Abstract*—In recent years, the performance of video super-resolution has improved significantly with the help of convolutional neural networks (CNN). Most recent works based on CNN use optical flow to handle video frames. They first compensate the motion and perform multi-frame super-resolution based on aligned frames. However, this 2-step approach has the drawback that the first step can be a bottleneck in overall performance. In this paper, we present a different approach to solving video super-resolution problem without any use of optical flow or motion compensation. We adopt recent advances in a recurrent neural network called long-short-term memory (LSTM) and residual network to deal with consecutive video frames effectively. Compared to the single-frame method, our recurrent model gives superior performance and shows more temporally coherent results.**

I. Introduction

Increasing a resolution of image or video while maintaining the visual quality is one of the most challenging and fundamental problems in computer vision. The main goal of super-resolution is to recover the details of the high-resolution (HR) image or video from its low-resolution (LR) counterpart. Recently, video super-resolution has attracted attention with the trend of devices equipped with high-resolution displays.

Especially, video super-resolution aims to generate the sequence of HR video frames given LR frames. Since the video frames share a lot of information between the frames, it is necessary to utilize temporal redundancy to design an efficient algorithm.

Most recent works tackle the video super-resolution problem by explicitly compensating local or global motions to make consecutive frames registered. They use optical flow in a variety of ways to estimate the motion. However, perfect prediction of optical flow is challenging and is still a research area. Thus relying on the optical flow as the first step of the algorithm is a dangerous way. The imperfection of the first stage can lead to a large performance loss in the overall algorithm.

In this paper, we present a more seamless way of dealing with video frames. Our algorithm excludes the use of any optical flow or motion compensation by introducing a recurrent module. The recurrent module operates convolutionally, thus making our CNN-based model specialized for video processing. We experimentally demonstrate the superiority of our model by comparing the conventional CNN with the proposed recurrent model.

II. Related Works

Recently, the powerful capability of deep convolutional neural networks (CNN) has led to dramatic improvements in various super-resolution(SR) problems. Since the work of Dong et al. [1], many super-resolution models based on deep CNN structure have been suggested. Kim et al. [2] first proposed to use residual connections in deep CNN. Using residual learning strategy significantly reduces the burden of network constructing trivial low-frequency part. Later, residual networks proposed by He et al. [3] have been adopted in super-resolution networks to enable the network much deeper while effective [4][13].

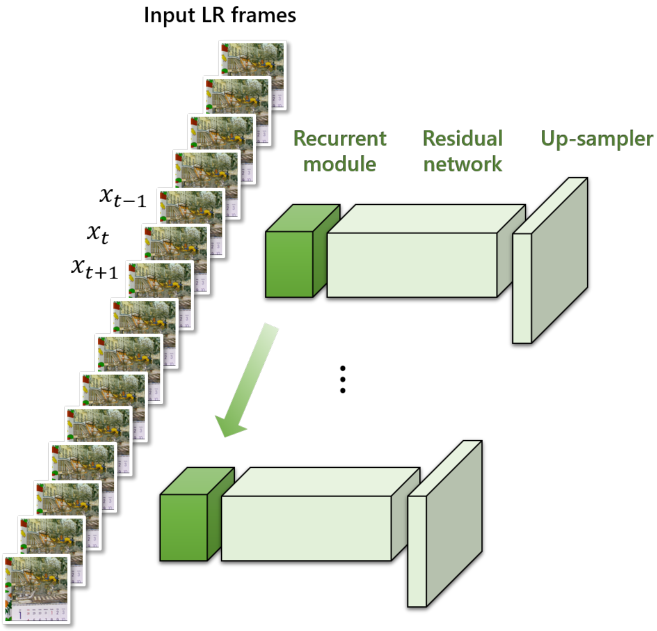


Fig. 1: The overall architecture of our network (DRRNet). We use EDSR[13] as a baseline structure. The recurrent module operates convolutionally, thus allows the model to process input frames effectively.

To benefit from temporal correlations between consecutive frames, video SR methods typically take more than a single image as input. Most works estimate the optical flow first using off-the-shelf algorithms [6][12][14][7], or by learning the optical flow network jointly [16][8]. Huang et al. [15] proposed to use a recurrent network for modeling long-term contextual information. Our model also works in a recurrent manner but uses much deeper architecture and residual learning techniques.

III. Proposed Method

A. Convolutional LSTM

To capture the long-term temporal video information, we use long-short-term memory (LSTM) [18] modules. LSTM is known to capture long-term dependencies well. To accommodate image processing, unlike regular LSTM cells, our LSTM module uses a convolution layer rather than fully connected ones. We replace every vector multiplication in LSTM with convolutions. Equation (1) describes the activation of the proposed convolutional Long-Short Term Memory cell.

$$\begin{aligned} i_n &= \sigma(x_n * w_{xi} + h_{n-1} * w_{hi} + b_i). \\ f_n &= \sigma(x_n * w_{xf} + h_{n-1} * w_{hf} + b_f). \\ \tilde{c_n} &= tanh(x_n * w_{x\tilde{c}} + h_{n-1} * w_{h\tilde{c}} + b_c). \\ c_n &= \tilde{c_n} \cdot i_n + c_{n-1} \cdot f_n. \\ o_n &= \sigma(x_n * w_{xo} + h_{n-1} * w_{ho} + b_o). \\ h_n &= o_n \cdot tanh(c_n). \end{aligned} \tag{1}$$

x_n and h_n denotes the input LR frame and output of the module. w and b are convolution kernels and bias, and $*$ denotes the convolution operation. i_n, f_n, c_n and o_n is the input, forget, cell, and output gate respectively. We use the sigmoid function and tanh activation, denoted by σ and *tanh* respectively.

B. Network Architecture

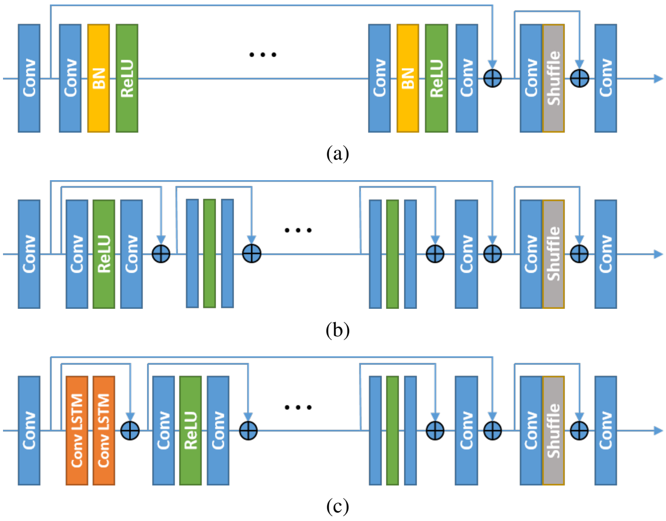
The proposed network (DRRNet) has a similar architecture with EDSR [13]. Please refer to Fig. 2 (c) for the overall structure of our proposed network. We use global skip connection and residual network architecture [3], and avoid using batch normalization layers. The first residual block is consists of two convolutional LSTM modules described in the previous section. Other residual blocks are consist of 2 convolution layers with single ReLU activation layer between them. All the convolutions have the same kernel size 3×3, and the number of channels is fixed to 64. The network upsamples the low-resolution feature map at the last step using deconvolution layer. For efficient implementation, we use the combination of convolution and pixel shuffle layers instead of deconvolution. They perform the same operation as demonstrated by Shi et al., [17]. We set the depth of the network by 10 residual blocks. Thus, the network has the depth of 23 convolution layers in total.

IV. Experimental Results

For training, we use the Berkeley Video Segmentation Benchmark (VSB100) [10]. VSB100 consists of 40 training sequences and 60 test sequences. We use 90 sequences for training and 10 sequences for validation. A mini-batch consists of 16 patches extracted randomly from consecutive frames. The patch size is set to 64. We use ADAM optimizer [11] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate is fixed to 10^{-4}. All the experiments are done in scale 4. We use the L1 loss as represented in (2) instead of the L2 loss, since L1 loss shows better performance. The loss function at time = T is defined as

$$loss_T = \frac{1}{\rho} \sum_{t=T-\rho+1}^{T} |y_t - \tilde{y_t}|. \tag{2}$$

where y_t and $\tilde{y_t}$ are target and output frames, repectively. ρ is the maximum depth of gradient back-propagation through time, and we set this value to 5. When training the convolutional LSTM modules, the maximum depth of gradient back-propagation through time is set to 5. The input frames x_t are generated by downsampling the target HR frames y_t using bicubic interpolation. We perform the experiments using a machine with 8-core 3.0 GHz CPU and a NVIDIA Titan X GPU. The implementation of our algorithm is based on the Torch7 framework.



(a)

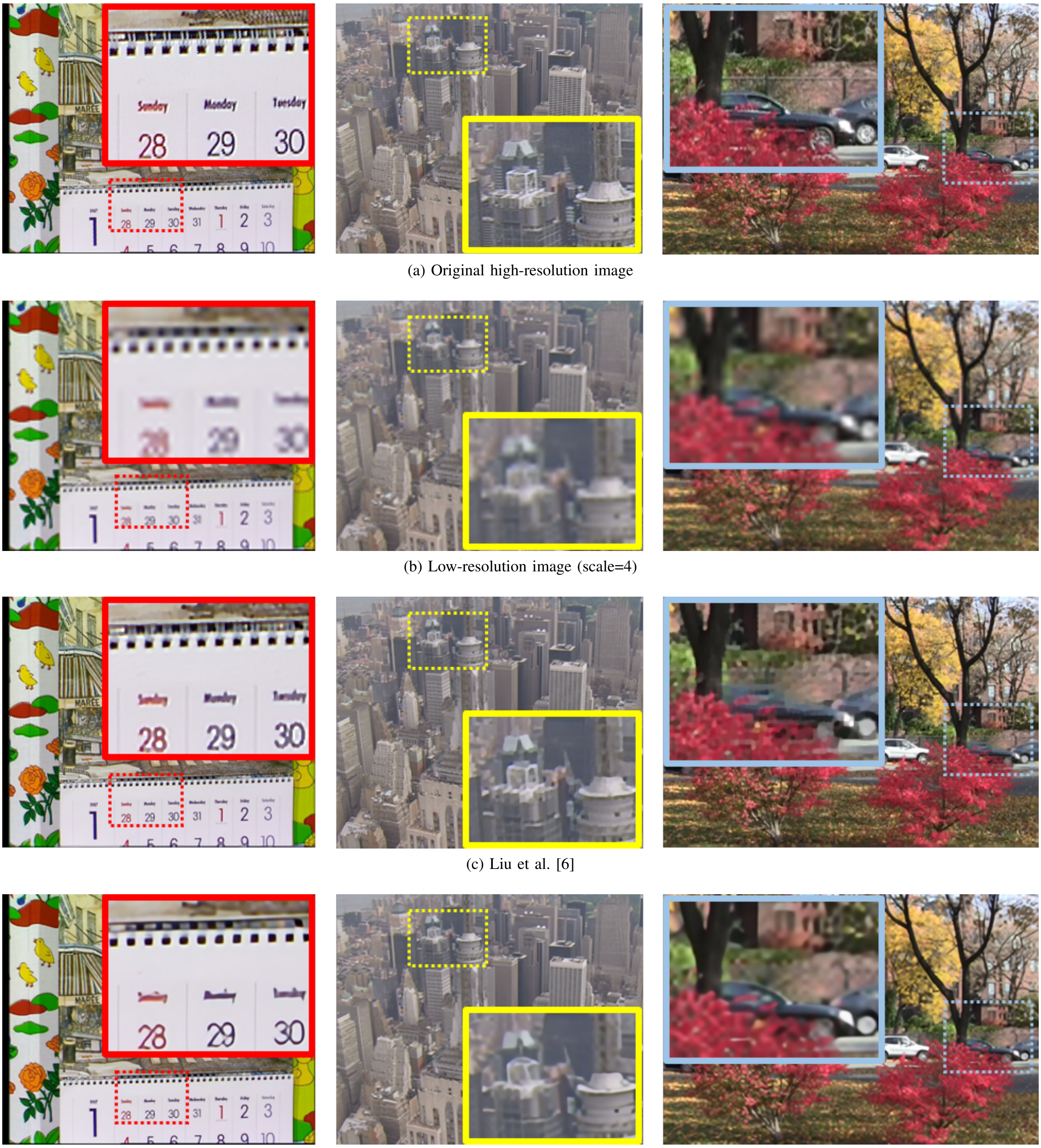
(b)

(c)

Fig. 2: Network architectures used for ablation study. (a) Modified VDSR. (b) Resnet. (c) Our DRRNet. The operation of the Convolutional LSTM module is described in (1).

To show the superiority of our model, we compare our DRRNet with several other network architectures. Fig. 2 depicts the network architectures used in the comparison. A CNN structure similar to VDSR [2] is the baseline model. We also compare the Resnet structure same as EDSR [13]. Evaluation of both Resnets with and without the use of batch normalization layers shows that it is better not to use them. TABLE I shows the comparative performance evaluation of these models on benchmark dataset [6]. Our model took 0.24 seconds to process each frame.

TABLE II shows the performance of SRCNN [1], VSRnet [12], ESPCN [5], and the proposed DRRNet on the *Videoset4* sequences (Calendar, Walk, Foliage, City) in [6]. We evaluate their performance in PSNR, which is the most widely-used metric. Only the Y channels of the results were used since some of them can only deal with a single channel. The PSNRs shown in the table are averaged over all sequences. Fig. 3



(a) Original high-resolution image

(b) Low-resolution image (scale=4)

(c) Liu et al. [6]

(d) Proposed algorithm

Fig. 3: Qualitative results for the benchmark dataset [6]. From left to right, '*calendar*', '*city*', '*foliage*' sequence.

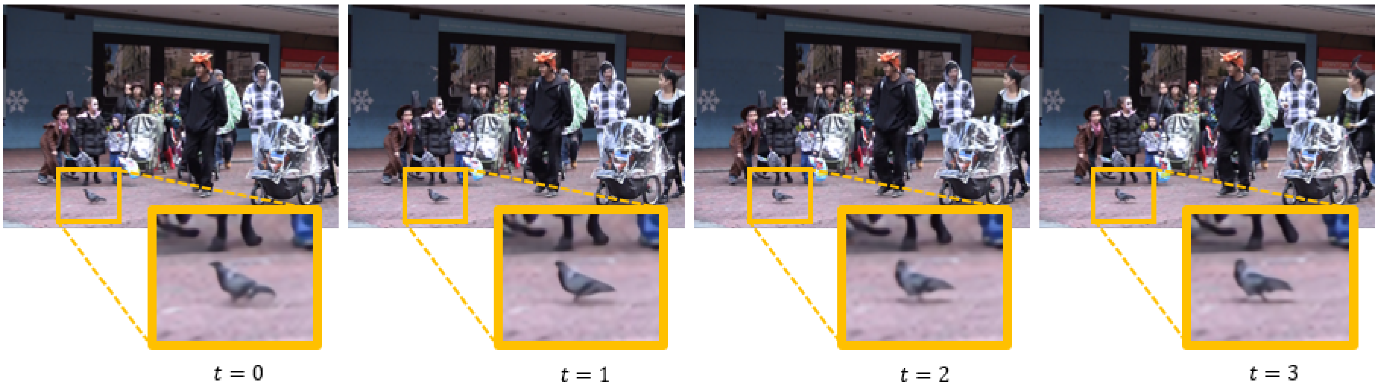


Fig. 4: The results of the proposed DRRNet for successive frames in '*walk*' sequence in [6]. The results show that our model produce stable and consistent SR images over time.

TABLE I: Model comparison on benchmark dataset [6]. Batch normalization does not help in the Resnet structure. Resnet structure combined with recurrent module improves performance.

	Resnet	No BN	Recurrent	PSNR
Simple				24.71 dB
Resnet w/ BN	✓			25.08 dB
Resnet w/o BN	✓	✓		25.10 dB
DRRNet	✓	✓	✓	25.15 dB

TABLE II: Quantitative results on benchmark dataset [6].

Scale		4
PSNR	Bicubic	23.82 dB
	SRCNN[1]	24.68 dB
	Bayesian[6]	25.06 dB
	VSRnet[12]	24.43 dB
	ESPCN[5]	25.06 dB
	DRRNet(ours)	25.15 dB

compares the result images of our algorithm with those of HR, bicubic interpolation, and Liu et al., [6]. Fig. 4 demonstrates that our network produces stable and consistent SR frames over time.

V. Conclusions

In this paper, we proposed a recurrent Resnet system for video super-resolution. By combining the advantages of deep CNN and RNN, our model can achieve high performance while omitting the complicated process of estimating object motion.

References

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2004.

[2] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image superresolution using very deep convolutional networks," In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2016.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2016.

[4] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A.Acosta et al., "Photo-realistic single image super-resolution using a generative adversarial network," In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2017.

[5] W. Shi, J. Caballero, H. Ferenc, T. Johannes, A. P. Aitken, R. Bishop et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2016.

[6] C. Liu and D. Sun, "A Bayesian approach to adaptive video super-resolution," In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2011.

[7] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," In Proceedings of International Conference of Computer Vision (ICCV), 2015.

[8] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang et al., "Real-time video super-resolution with spatio-temporal networks and motion compensation," In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2017.

[9] N. Vinod, and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," In Proceedings of the 27th International Conference on Machine Learning (ICML), 2010.

[10] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, B. Schiele, "A unified video segmentation benchmark: Annotation, metrics and analysis," In Proceedings of International Conference on Computer Vision (ICCV), 2013.

[11] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," In Proceedings of International Conference on Learning Representations (ICLR), 2015.

[12] A. Kappeler, S. Yoo, Q.Dai and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," IEEE Transactions on Computational Imaging, 2016.

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," In Proceedings of Computer Vision and Pattern Recognition Workshop (CVPRW), 2017.

[14] D. Li and Z. Wang, "Video Super-Resolution via Motion Compensation and Deep Residual Learning," IEEE Transactions on Computational Imaging, 2017.

[15] Y. Huang, W. Wang, and L. Wang, "Bidirectional Recurrent Convolutional Networks for Multi-Frame Super-Resolution," In Proceedings of Neural Information Processing Systems (NIPS), 2016.

[16] O. Makansi, E. Ilg, and T. Brox "End-to-End Learning of Video Super-Resolution with Motion Compensation," In Proceedings of German Conference on Pattern Recognition, 2017.

[17] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig et al., "Is the deconvolution layer the same as a convolutional layer?," arXiv preprint arXiv:1609.07009, 2016.

[18] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural Computation, 1997.