VIDEO SUPER-RESOLUTION USING MOTION COMPENSATION AND RESIDUAL BIDIRECTIONAL RECURRENT CONVOLUTIONAL NETWORK

Dingyi Li[⋆] *Yu Liu*[‡] *Zengfu Wang*[⋆†]

[⋆] Department of Automation, University of Science and Technology of China, Hefei, China
[‡] Department of Biomedical Engineering, Hefei University of Technology, Hefei, China
[†] Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China

lidingyi@mail.ustc.edu.cn, yuliu@hfut.edu.cn, zfwang@ustc.edu.cn

ABSTRACT

Video super-resolution (SR) aims at restoring finer details and enhancing visual experience. In this paper, we propose a novel method named residual recurrent convolutional network (RRCN) for video SR. In our method, motion compensation and bidirectional residual convolutional network are combined to model the spatial and temporal non-linear mappings. To leverage sufficient amount of temporal information, we employ motion compensation, bidirectional recurrent convolutional layers and late fusion in of our network. We also apply residual connections in our recurrent structure for more accurate SR. Experimental results demonstrate the superiority of the proposed method over state-of-the-art single-image and multi-frame based SR approaches in terms of both quantitative assessment and visual quality.

Index Terms— Video super-resolution, recurrent neural networks (RNNs), convolutional neural networks (CNNs), deep residual learning

1. INTRODUCTION

Video super-resolution (SR) aims to determine the values of all the pixels in high-resolution (HR) frames from low-resolution (LR) observations. Video SR can provide visually appealing videos and help to make full use of the recently sprung up HR display devices.

SR methods can be classified into two categories: single-image based and multi-frame based [1]. Single-image SR algorithms include interpolation-based and example-based methods. Interpolation-based SR methods are usually of high computational efficiency, but tend to generate over-smoothed or jagged images. Example-based methods learn the non-linear mappings from LR patches to HR patches using external and/or internal examples. With sufficient information in the external dataset or in the same input image across different scales [2, 3], example-based methods provide visually appealing results. Traditional example-based methods include nearest neighbor [4], neighbor embedding [5], sparse coding [6], anchored neighborhood regression [7] and random forest [8] based ones. With the pioneering work that uses convolutional neural network (CNN) for single-image SR by Dong *et al.* [9], deep learning has been widely used in single-image SR [10–15]. He *et al.* proposed deep residual learning [16] to tackle the degradation problems for very deep CNNs. Deep residual learning has also been introduced in single-image SR and achieved accurate restoration and perceptually pleasing results [11, 12, 14, 15]. With residual learning, these SR nets are of better representation abilities.

Previous multi-frame SR methods are mainly reconstruction-based and leverage inter-frame consistencies. Most of them are based on a bayesian framework and adopt optical flow techniques for sub-pixel accuracy motion estimation [17–19]. These methods can ensure high fidelity if small global motions are in existence. However, they usually lose effectiveness when scale factors are large or large motions occur.

Recently, there have been some work on combining the representation ability of deep learning and inter-frame temporal consistences to enhance both visual quality and fidelity [20–23]. Huang *et al.* [20] proposed an end-to-end bidirectional recurrent convolutional network (BRCN) for fast video SR. BRCN may generate visual artifacts on the edges since no explicit registration is conducted. Liao *et al.* [21] and Kappler *et al.* [22] combined motion compensation with CNNs in a two-step pipeline. They tackled large and complex motions using deep-draft ensembles [21] and adaptive motion compensation [22], respectively. However, these techniques may also miss some useful inter-frame information. Li and Wang [23] proposed a new net named MCResNet for video SR which is able to handle large and complex motions. They utilized early fusion for multiple frames.

In this paper, we propose a novel method named residual recurrent convolutional network (RRCN) for video SR. To the best of our knowledge, this is the first time motion compensated frames are regarded as the inputs of recurrent convolutional nets for video SR. And it is also the first time recurrent convolutional layers are utilized to reconstruct the residual images rather than the output images in video SR. In order to model the temporal dependencies better, bidirectional recurrent convolutional layers and late fusion are employed on the motion compensated frames. We utilize skip-layer connections in our recurrent structure to handle large and complex motions to achieve more accurate SR. Our method has demonstrated state-of-the-art results on challenging video datasets.

The main contributions of this work are threefold:

- We propose a novel residual recurrent convolutional net for video SR.
- We employ motion compensation, bidirectional recurrent convolutional layers and late fusion to leverage temporal dependencies.

This work is supported by the National Natural Science Foundation of China (no. 61472393). Corresponding author: Z. Wang.

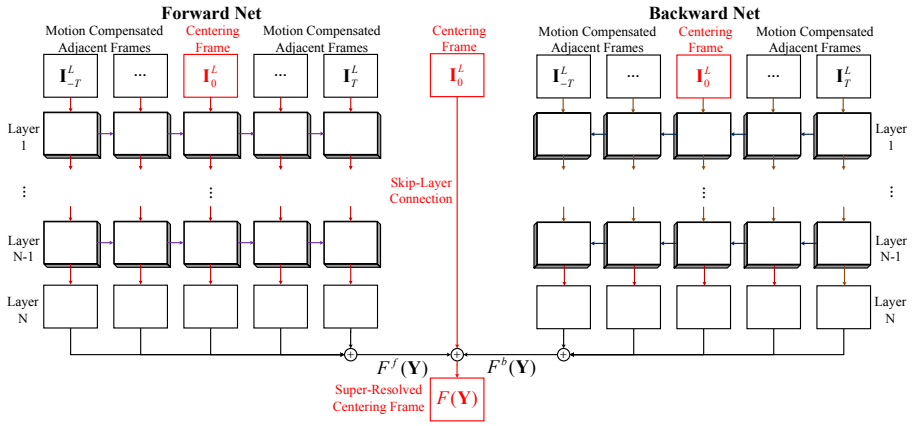


Fig. 1. The proposed network structure for video SR

- In our recurrent network, we adopt residual connections between the centering input frame and the centering output frame for better performance.

The remainder of the paper is organized as follows. In Section 2 we illustrate the details of our video SR method RRCN. Quantitative and qualitative results of our method and the state-of-the-art SR methods are provided in Section 3. Section 4 draws a conclusion.

2. THE PROPOSED METHOD

2.1. Overview

Our method is a two-step approach. The first step is motion compensation and the later is our RRCN. Different from [20], we perform recurrent convolutional operations on motion compensated frames instead of on raw inputs. A recurrent structure is employed to predict not the HR images but the residual images. We restore one frame from multiple frames by late fusion.

Following [22], we first use the bicubic interpolation to upsample LR input frames with specific scaling factors. Then we use the combined local-global with total variational (CLG-TV) optical flow approach [24] for motion estimation between the centering frame and each adjacent frame. We regard the centering frame as the reference frame to compensate each adjacent frame. We feed the centering frame and the motion compensated frames into our RRCN for video SR.

2.2. Our RRCN

BRCN [20] is an end-to-end model to learn non-linear mappings from multiple input LR frames to multiple output HR frames. However, we find that for deep learning based video SR it is better to use multiple frames to restore one centering frame rather than super-resolving multiple-frames simultaneously. We also observe that although motion estimation is time-consuming, it is crucial to get more accurate restoration results and reduce visual artifacts. So our deep neural network starts with the motion compensated input frames. We add skip connection between the input centering frame and the output centering frame to preserve low-frequency information and restore finer details. We are only predicting the residual image. We illustrate our network structure in Fig. 1. Suppose we use $2T+1$ input frames and our RRCN consists of N layers. Let $\mathbf{Y} = \{\mathbf{I}^L_{-T}, \mathbf{I}^L_{-T+1}, \cdots, \mathbf{I}^L_0, \cdots, \mathbf{I}^L_{T-1}, \mathbf{I}^L_T\}$ to be the set of all input frames where $\mathbf{I}^L_t$ is the tth input low-resolution observation. The first $(N-1)$ layers are of feed-forward convolutional layers and recurrent convolutional layers. The equations of the first $(N-1)$ layers are

$$F^f_i(\mathbf{I}^L_t) = max(0, W^f_i * F^f_{i-1}(\mathbf{I}^L_t) + V^f_i * F^f_i(\mathbf{I}^L_{t-1}) + B^f_i) \quad (1)$$

and

$$F^b_i(\mathbf{I}^L_t) = max(0, W^b_i * F^b_{i-1}(\mathbf{I}^L_t) + V^b_i * F^b_i(\mathbf{I}^L_{t-1}) + B^b_i) \quad (2)$$

for the forward and backward network respectively. Where i is the layer number and t is the input frame number. F^f_i (or F^b_i) is the output of the ith layer in the forward (or backward) net. W^f_i (or W^b_i) and V^f_i (or V^b_i) represent the filters of the ith feed-forward convolutional layer and recurrent convolutional layer in the forward (or backward) net, respectively. Note that we let $F^f_0(\mathbf{I}^L_t) = F^b_0(\mathbf{I}^L_t) = \mathbf{I}^L_t$. All of the first $(N-1)$ feed-forward convolutional layers have 32 feature maps and a small filter size of 3×3. All recurrent convolutional layers have 32 feature maps and 1×1 filters. B^f_i and B^b_i are the biases for the ith forward and backward layer respectively. $*$ denotes a convolutional operation. Here we choose the rectified linear unit (ReLU) [25] as the activation function.

The equations of the last layer are

$$F^f_N(\mathbf{I}^L_t) = W^f_N * F^f_{N-1}(\mathbf{I}^L_t) + B^f_N \quad (3)$$

and

$$F^b_N(\mathbf{I}^L_t) = W^b_N * F^b_{N-1}(\mathbf{I}^L_t) + B^b_N \quad (4)$$

where W^f_N (or W^b_N) and B^f_N (or B^b_N) represent the filters and the biases respectively. 3×3 filters are used for the last feed-forward convolutional layer. The output of each frame is a single residual feature map.

We utilize late fusion at the top of our network. We find that using only the output centering residual images leads to plenty of noise. So we instead utilize element-wise sum of all residual images

$$F^f(\mathbf{Y}) = \sum_{t=-T}^{T} F^f_N(\mathbf{I}^L_t) \quad (5)$$

Table 1. Average PSNR and SSIM values for the *Myanmar* testing dataset.

Scale	Metric	Single-image based approaches				Multi-frame based approaches			
		Bicubic	SRCNN [26]	VDSR [11]	DRCN [12]	Bayes [18]	Enhancer [27]	VSRnet [22]	RRCN
×2	PSNR	34.59	37.79	38.56	38.43	35.56	35.94	38.48	**40.00**
	SSIM	0.9458	0.9640	0.9671	0.9670	0.9515	0.9588	0.9679	**0.9789**
×3	PSNR	31.59	33.88	34.64	34.71	32.20	32.50	34.42	**35.23**
	SSIM	0.8957	0.9198	0.9257	0.9262	0.9203	0.9099	0.9247	**0.9421**
×4	PSNR	29.53	31.26	32.29	**32.32**	30.68	30.23	31.85	32.28
	SSIM	0.8526	0.8777	0.8873	0.8873	0.8895	0.8681	0.8834	**0.9029**

Table 2. Average PSNR and SSIM values for the *Videoset4* dataset.

Scale	Metric	Single-image based approaches				Multi-frame based approaches			
		Bicubic	SRCNN [26]	VDSR [11]	DRCN [12]	Bayes [18]	Enhancer [27]	VSRnet [22]	RRCN
×2	PSNR	28.43	30.70	31.44	31.68	29.69	30.40	31.30	**32.44**
	SSIM	0.8676	0.9172	0.9257	0.9269	0.9055	0.9151	0.9278	**0.9464**
×3	PSNR	25.28	26.51	26.82	26.99	25.82	26.34	26.79	**27.65**
	SSIM	0.7329	0.7933	0.8089	0.8122	0.8323	0.7948	0.8098	**0.8557**
×4	PSNR	23.79	24.69	24.98	25.03	25.06	24.55	24.84	**25.51**
	SSIM	0.6332	0.6918	0.7119	0.7141	0.7466	0.6877	0.7049	**0.7612**

and

$$F^b(\mathbf{Y}) = \sum_{t=-T}^{T} F_N^b(\mathbf{I}_t^L) \tag{6}$$

where $F^f(\mathbf{Y})$ and $F^b(\mathbf{Y})$ are the final outputs of the forward net and the backward net respectively. Note that by using this sum operation, our recurrent network differs from traditional recurrent nets where the input frame numbers are unconstrained. For our network, the input frame numbers are fixed to $2T + 1$.

Since the upsampled LR image is close to the ground truth HR image, it is better for the network to learn only the residual components. By adding skip connection between the input centering frame and the output forward and output backward centering residual frame, the low-frequency information in the LR frame are maintained and the high-frequency details can be better restored by our recurrent convolutional net. The equation for the final output of our network is

$$F(\mathbf{Y}) = \mathbf{I}_0^L + F^f(\mathbf{Y}) + F^b(\mathbf{Y}) \tag{7}$$

where $F(\mathbf{Y})$ represents the output of our entire RRCN.

2.3. Network Training

We utilize mean squared error (MSE) as our loss function:

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^{n} ||F(\mathbf{Y}_i, \Theta) - \mathbf{X}_i||_2^2 \tag{8}$$

where Θ are the network parameters, n is the number of training samples in a mini-batch and $\mathbf{X}_i$ is the ith ground truth HR centering frame. We employ RMSProp [28] for network optimization. We firstly train a net for $\times 2$. Then we use the net parameters of $\times 2$ to initialize nets for $\times 3$ and $\times 4$ in sequence for faster training.

3. EXPERIMENTS

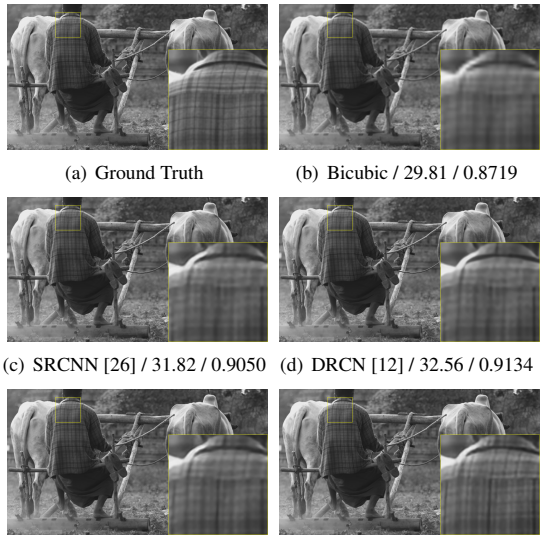
3.1. Implementation Details

We utilize 53 scenes in the *Myanmar* [29] video dataset for training and the rest 6 scenes for testing as in [22]. All frames in this dataset are downsampled to 960×540. Note that we only use this small video dataset while in [22] the network is finetuned using the super-resolution CNN (SRCNN) model [26] which is trained on the huge dataset ImageNet [30]. We also use another widely-used dataset *VideoSet4* [18, 22] for performance evaluation. The *VideoSet4* dataset consists of four videos: *calender*, *city*, *foliage* and *walk*.

We adopt data augmentation by changing spatial and temporal scales in the training datasets. Each input frame is firstly downsampled and then upsampled using the bicubic interpolation. We employ motion compensation between the centering frame and each adjacent frame. We use motion compensated frames as the inputs of our network. 5 input frames and 10 layers are used in all of our experiments. So we have $T = 2$ and $N = 10$.

3.2. Quantitative and Qualitative Evaluations

We compare our method with state-of-the-art single-image based and multi-frame based SR algorithms. Single-image SR methods include the bicubic interpolation (Bicubic), SR convolutional neural network (SRCNN) [26], very deep SR (VDSR) [11], deep recursive convolutional network (DRCN) [12]. Multi-frame based methods consist of Bayes [18], a commercial software "Video Enhancer" [27] (related to [17], a bayesian method) and VSRnet [22]. All methods are performed on the Y channel of the LR images in YCbCr space using Bicubic with different scale factors. Other channels are down-

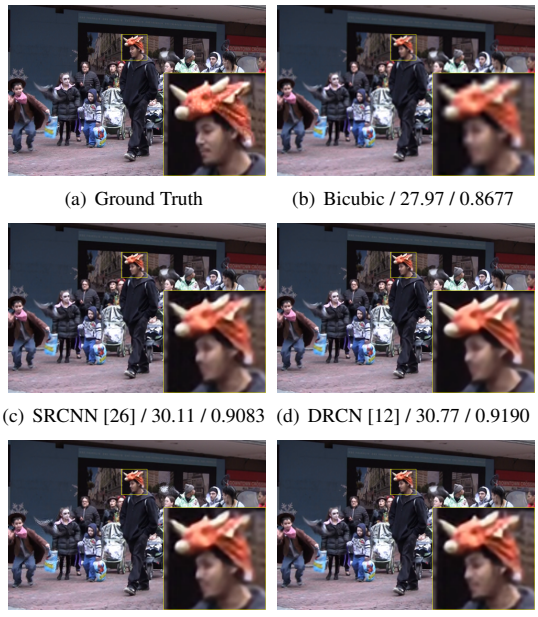


(a) Ground Truth (b) Bicubic / 29.81 / 0.8719

(c) SRCNN [26] / 31.82 / 0.9050 (d) DRCN [12] / 32.56 / 0.9134

(e) VSRnet [22] / 32.16 / 0.9103 (f) RRCN / **32.88** / **0.9279**

Fig. 2. SR results and PSNR, SSIM values for one frame in the *Myanmar* testing dataset for $\times 3$ using different methods.



(a) Ground Truth (b) Bicubic / 27.97 / 0.8677

(c) SRCNN [26] / 30.11 / 0.9083 (d) DRCN [12] / 30.77 / 0.9190

(e) VSRnet [22] / 30.01 / 0.9072 (f) RRCN / **31.23** / **0.9343**

Fig. 3. SR results and PSNR, SSIM values for a *walk* video frame in the *Videoset4* dataset for $\times 3$ using different methods.

sampled and later upsampled using Bicubic. All multi-frame based methods use 5 input frames to super-resolve 1 centering frame. CNN based methods [11, 12, 22, 26] were trained by the authors using stochastic gradient descent (SGD). [18] and [27] need no training. We use peak pixel-to-noise ratio (PSNR) and structural similarity (SSIM) [31] as our objective quality assessments. When evaluating, we eliminate 8 pixels on each border as in [22]. Detailed results are illustrated in Table 1 and 2. Our method outperforms other methods in SSIM values and obtains the highest PSNR values except for $\times 4$ in the *Myanmar* testing dataset. Our RRCN gets more PSNR gains for smaller scaling factors since in these cases the motion estimation is much more accurate and provides more sufficient complementary information. For an output image of 960×540, our method takes about 21.93 seconds on motion compensation with an Intel i7-6700K CPU of 4.0 GHz and about 1.39 seconds on our RRCN with the same CPU and a Nvidia Titan X GPU. The running time of our network is competitive to other CNN based [11, 12, 22, 26] and BRCN based [20] SR approaches.

Fig. 1 and 2 are some of the SR results. In Fig. 1, our method restores finer texture details of the stripes on the man's shirt. Other methods provide over-smoothed results and are unable to restore the thick vertical black line in the middle. As illustrated in Fig. 2, the facial elements such as eye and mouth are better restored by our method. We also find that VSRnet with motion compensation [22] generates artifacts on the fast flying pigeon because of inaccurate motion estimation. We use the same motion estimation method and find that our network is able to handle large and complex motions adaptively. More results are provided online[1].

Table 3. Average PSNR values for the *VideoSet4* testing dataset for $\times 3$ with different settings, "MC" represents motion compensation and "Res" means residual connections.

Setting	PSNR	SSIM
Without MC, without Res	26.81	0.8183
Without MC, with Res	27.12	0.8322
With MC, without Res	27.36	0.8455
With MC, with Res	**27.65**	**0.8557**

3.3. More Analysis

In Table 3 we show that without motion compensation or residual connections, the PSNR values drop significantly. We also observe serve border artifacts on the SR results without motion compensation. So both motion compensation and residual connection are crucial for the best performance. The "no MC" version in Table 3 can be regarded as a rapid version of our method since it needs no explicit motion estimation which is time-consuming.

4. CONCLUSION

In this paper, we have proposed a novel method RRCN for video SR. Our method models temporal dependencies sufficiently by using motion compensation, bidirectional convolutional layers and late fusion. We employ deep residual learning in our recurrent structure to further improve the representation ability of our network and handle large and complex motions. Our method has demonstrated superior performance over state-of-the-art SR methods.

[1] https://drive.google.com/drive/folders/0B0xZuPOY8Cd2NEJvRWdwTzR4NDA?usp=sharing

5. REFERENCES

[1] K. Nasrollahi and T. Moeslund, "Super-resolution: a comprehensive survey," *Mach. Vis. & Appl.*, vol. 25, no. 6, pp. 1423–1468, 2014.

[2] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 349–356.

[3] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5197–5206.

[4] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. and Appl.*, vol. 22, no. 2, pp. 56–65, 2002.

[5] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, vol. 1, pp. 275–282.

[6] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[7] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. IEEE Asian Conf. Comput. Vis.*, 2014, pp. 111–126.

[8] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3791–3799.

[9] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[10] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 370–378.

[11] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1646–1654.

[12] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1637–1645.

[13] X. J. Mao, C. Shen, and Y. B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2802–2810.

[14] J. Johnson, A. Alahi, and F. F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[15] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *arXiv preprint arXiv:1609.04802*, 2016.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[17] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1327–1344, 2004.

[18] C. Liu and D. Sun, "On bayesian adaptive video super resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 346–360, 2014.

[19] Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, and E. Wu, "Handling motion blur in multi-frame super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5224–5232.

[20] Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 235–243.

[21] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 531–539.

[22] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Trans. Comput. Imag.*, vol. 2, no. 2, pp. 109–122, 2016.

[23] D. Li and Z. Wang, "Video super-resolution via motion compensation and deep residual learning," *IEEE Trans. Comput. Imag.*, pp. 1–15, 2017.

[24] M. Drulea and S. Nedevschi, "Total variation regularization of local-global optical flow," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2011, pp. 318–323.

[25] B. V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proc. Int. Conf. Mach. Learn.*, pp. 807–814, 2010.

[26] C. Dong, K. He C. C. Loy, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2016.

[27] "http://www.infognition.com/videoenhancer," 2010.

[28] N. S. Geoffrey Hinton and K. Swersky, "Neural networks for machine learning lecture 6a: Overview of mini-batch gradient descent," .

[29] "http://www.harmonicinc.com/resources/videos/4k-video-clip-center," 2014.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, M. Bernstein A. Khosla, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.