Postprint: Video Image Super-Resolution Reconstruction Method Based on Convolutional Neural Networks

Authors: Liucun, Li Yuanxiang, Zhou Yongjun, Luo Jianhua

Date: 2018-05-02T00:00:00+00:00

Abstract

To further enhance the effectiveness of video image super-resolution reconstruction, this research leverages the characteristics of convolutional neural networks for spatial resolution reconstruction of video images and proposes a video image reconstruction model based on convolutional neural networks. A pre-training strategy is adopted for initializing the reconstruction model parameters, while training is conducted simultaneously on the spatial and temporal dimensions of multi-frame video images to extract features describing primary motion information for learning, fully utilizing the information complementarity between video frames to reconstruct intermediate frames. To address motion blur in inter-frame images, adaptive motion compensation is employed for processing, and channels are optimized to output high-resolution reconstructed images. Experiments demonstrate that the reconstructed video images achieve significant improvements in average objective evaluation metrics (PSNR +0.4 dB / SSIM +0.02) and effectively reduce edge blur phenomena in subjective visual quality. Compared with other traditional algorithms, notable improvements are observed in both objective metrics and subjective visual effects of image evaluation, providing a novel architecture based on convolutional neural networks for video image super-resolution reconstruction and offering insights for further exploration of deep learning-based video image super-resolution reconstruction methods.

Full Text

Preamble

Video Image Super-Resolution Reconstruction Method Based on Convolutional Neural Network

Liu Cun, Li Yuanxiang, Zhou Yongjun, Luo Jianhua

(School of Aeronautics & Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: To further enhance the performance of video image super-resolution reconstruction, this paper proposes a novel video reconstruction model based on convolutional neural networks (CNNs) that leverages the characteristics of CNNs for spatial resolution enhancement of video images. The model employs a pre-training strategy to initialize reconstruction parameters and simultaneously trains on both spatial and temporal dimensions of multi-frame video sequences. It extracts features that describe primary motion information for learning, fully utilizing complementary information across video frames to reconstruct intermediate frames. To address motion blur between frames, adaptive motion compensation is adopted, and channel optimization is applied to the output to obtain high-resolution reconstructed images. Experiments demonstrate significant improvements in average objective evaluation metrics (PSNR +0.4 dB / SSIM +0.02) and effectively reduce edge blurring in subjective visual quality. Compared with traditional algorithms, the proposed method achieves clear improvements in both objective metrics and subjective visual effects, providing a novel CNN-based architecture for video super-resolution reconstruction and offering insights for further exploration of deep learning-based video super-resolution methods.

Keywords: video; super-resolution reconstruction; convolutional neural network; deep learning

0 Introduction

Image super-resolution reconstruction is the process of obtaining a corresponding high-resolution image from low-resolution images or video sequences. This technique has widespread applications in medical imaging, aerospace, electronic surveillance, and other fields [1]. With the increasing popularity of next-generation ultra-high-definition video (3840×2048), most video content faces numerous challenges during acquisition, transmission, and storage. Consequently, video reconstruction algorithms are needed to generate ultra-high-definition content from full HD (1920×1080) or lower-resolution sources.

Current image super-resolution reconstruction methods can be divided into two categories: model-based reconstruction methods and learning-based reconstruction methods. Model-based approaches model low-resolution images as downsampled versions of high-resolution images with random noise, introducing regularization terms to constrain the model during high-resolution image recovery from low-resolution inputs [2]. Within a Bayesian framework, prior knowledge determining image smoothness is introduced to obtain higher-quality reconstructed images. For instance, Babacan et al. [3] utilized a Bayesian framework to reconstruct high-resolution images from multi-frame rotated and translated

low-resolution images, while Belekos et al. [2] and Liu et al. [4] also employed Bayesian frameworks to derive algorithms capable of handling complex object motion and real-scene video sequences.

The core idea of learning-based reconstruction algorithms is to learn the mapping relationship between high-resolution images and their corresponding low-resolution counterparts. In these methods, a dictionary is jointly trained from high-resolution and low-resolution image patches, where each low-resolution patch can be represented as a sparse linear combination of atoms from the corresponding low-resolution dictionary, with coefficients obtained through training serving as weights [5]. The dictionary and its weight coefficients can be obtained using standard sparse coding techniques such as K-SVD [6], after which high-resolution patches are reconstructed by finding sparse coefficients for corresponding low-resolution patches and applying them to the high-resolution dictionary. In dictionary-based reconstruction methods, natural image patches can be sparsely represented as linear combinations of learned dictionary atoms or blocks. Yang et al. [5] first proposed using two coupled dictionaries to learn the nonlinear mapping between low-resolution and high-resolution images. Song et al. [7] introduced a dictionary learning method for video super-resolution—real-time dictionary learning—which assumes that sparse keyframes exist in high-resolution images. Learning-based image reconstruction methods typically learn representations of image patches to reconstruct corresponding high-resolution images [8]. To avoid obvious edge effects in reconstructed images, algorithms often use overlapping patches, resulting in considerable computational overhead that affects efficiency. Timofte et al. [9] proposed replacing a single large incomplete dictionary with several smaller complete dictionaries to reduce computational costs in the sparse coding step while maintaining reconstruction accuracy and further improving algorithm speed. Schulter et al. [10] proposed training random forest models instead of directly training coupled dictionaries mapping low-resolution to high-resolution patches. Glasner et al. [11] did not learn dictionaries from sample images but instead created a set of low-resolution images with different scaling factors, then matched patches from low-resolution images with their scaled versions to reconstruct corresponding high-resolution patches using their high-resolution counterparts.

Traditional single-image super-resolution reconstruction primarily focuses on analyzing the image degradation process to obtain more modeling information during low-to-high resolution reconstruction, or fully utilizes image self-similarity and related transform domain analysis methods to recover more high-frequency information during super-resolution reconstruction. For video image super-resolution reconstruction, however, additional challenges must be addressed, including information loss, spatial shifts, and flipping between video frames, or redundancy of inter-frame information. The reconstruction process must effectively solve problems such as inter-frame information matching, preprocessing for multi-frame image reconstruction, orientation estimation, motion estimation, and motion blur, making parameter estimation for degradation models and reconstruction more difficult.

With the tremendous success of deep learning and convolutional neural networks in image recognition, a new generation of image reconstruction algorithms based on deep neural networks has begun to demonstrate excellent performance [15]. Deep neural networks can learn from and process large training databases such as ImageNet, and convolutional neural network training can be efficiently implemented through parallelization using GPU-accelerated computation. Moreover, once the network model is trained, the image super-resolution reconstruction process becomes a pure feedforward process, making CNN-based reconstruction algorithms superior in terms of temporal performance [16]. Dong et al. [17] pointed out that each step of dictionary learning-based reconstruction algorithms can be redefined as a layer in a deep neural network, where representing an image patch with a dictionary of atoms—i.e., applying filters with kernel size on the input image—can be implemented as a convolutional layer. Consequently, they proposed a convolutional neural network model that directly learns the nonlinear mapping from low-resolution to high-resolution images using a CNN structure with two hidden layers and one output layer. Wang et al. [18] introduced a patch-based method where convolutional autoencoders are applied to 33×33 pixel patches, using normalized low/high-resolution patches with subtracted means for training the reconstruction model. Training patches are clustered according to their similarity, and for each cluster, self-similar patches and each sub-model are fine-tuned to fully utilize the two-dimensional data structure of images. Additionally, training data is augmented with translation, rotation, and different scaling factors to enable the model to more effectively learn visually meaningful features. Cui et al. [19] proposed an algorithm that gradually increases the resolution of low-resolution images to the desired resolution using a cascade of collaborative local autoencoders (CLA). At each layer of the cascade, non-local self-similarity search (NLSS) is performed to reconstruct high-frequency details and textures of the image, with the resulting image then processed by an autoencoder to eliminate structural distortion and errors introduced by the NLSS step. This algorithm uses 7×7 pixel overlapping blocks, resulting in very large computational overhead. Cheng et al. [20] introduced a patch-based video reconstruction algorithm using fully connected network layers, where the network includes hidden and output layers and uses 5 consecutive low-resolution frames to reconstruct one central high-resolution frame. The video is processed in patches, with network input being 5×5×5 patches and outputting reconstructed patches from high-resolution images, finding corresponding patches through reference frame and neighboring frame matching. Liao et al. [21] proposed a similar method involving multi-frame motion compensation and frame combination using convolutional neural networks. First, two motion compensation algorithms with different parameter settings are used to calculate initial reconstruction results to handle motion compensation errors; then all initial versions of reconstructed images are combined using a convolutional neural network. However, this algorithm requires computing multiple motion compensation quantities for each frame, resulting in very large computational overhead.

Bayesian framework-based video reconstruction methods employ more complex optical flow algorithms or hierarchical block matching methods to estimate motion fields to handle real-scene videos with more complex motion patterns [12]. Ma et al. [13] extended Bayesian framework-based work to handle videos with motion blur and introduced the concept of temporal relative degree to remove severely blurred pixels, thereby obtaining better experimental results. Takeda et al. [14] proposed an alternative approach to conventional motion estimation and image reconstruction schemes, directly utilizing 3-D iterative steering kernel regression instead of simple motion estimation. Videos can be segmented and processed using overlapping 3D cubes (in both temporal and spatial dimensions), and high-resolution images are then reconstructed by approximating pixels in the cubes using 3D Taylor series.

In summary, most video reconstruction algorithms rely on accurate motion estimation between low-resolution video frames. Many algorithms require simultaneous estimation of dense motion fields and computation of motion vectors, which not only incurs enormous computational overhead but also requires adding considerably complex preprocessing steps before super-resolution reconstruction.

1 Video Reconstruction Method Based on Convolutional Neural Network

This paper proposes a novel video reconstruction model based on convolutional neural networks that uses multiple low-resolution video frames as input. Before input, an adaptive motion compensation mechanism is introduced to handle fast-moving objects in consecutive video frames and the resulting motion blur phenomenon. Channel optimization is also performed on the output layer to reconstruct a high-resolution output frame image. This paper also compares different convolutional network model structures and adopts an image pre-training strategy, using the obtained weight coefficients to initialize training parameters in the video reconstruction model, thereby improving overall model performance in terms of both reconstruction accuracy and speed.

1.1 Video Reconstruction Model

Learning-based reconstruction methods have shown that using neighboring frames is helpful for video reconstruction [22]. In the video reconstruction process, modeling and estimating inter-frame motion can obtain additional information through sub-pixel motion between frames. If the training process includes multiple video frames, learning-based methods can also obtain additional information conveyed by inter-frame differences.

The video reconstruction model architecture ([Figure 1: see original paper]) mainly includes a motion compensation module and three convolutional layers (L_1, L_2, and L_3). This paper combines adjacent frames (previous frame y_{t-1}, current frame y_t, and next frame y_{t+1}) as the input frame structure throughout

the reconstruction process. To use multiple forward and backward frames, the model architecture can be extended with additional branches. The framework size for single-frame input is $M \times N \times 1$, where M and N are the width and height of the input image frame, respectively. For the proposed model' s input architecture, the three input frames are concatenated along the first dimension before applying the first convolutional layer L_1, resulting in a new input data structure of $M \times N \times 3$ dimensions. Similarly, the model can also perform frame concatenation after convolutional layer L_1 and use it as input for L_2, with image data dimensions and filter sizes of the video reconstruction model adjusted accordingly. Because there are 3 input frames, the new filter size of the model is $f_1 \times f_1 \times 3 \times n_1$, where n_1 represents the number of kernels in layer 1. The filter size of convolutional layer L_1 is expanded to $f_1 \times f_1 \times 3$, and the filter size of convolutional layer L_2 is $f_2 \times f_2 \times n_1 \times n_2$.

In experiments, this paper initializes the filter weights and biases of the video reconstruction model with the same initial parameter configuration as the image pre-training model. The advantage is that the algorithm is not patch-based and ensures time efficiency of the algorithm. Here, X represents the input low-resolution image, and Y represents the output high-resolution image. The pre-training model consists of three convolutional layers, with two hidden layers L_1 and L_2 followed by rectified linear units (ReLU). The filters of convolutional layer L_1 are composed of n_1 kernels of size $f_1 \times f_1 \times 1$, where f_1 denotes the kernel size and n_1 denotes the number of kernels. The filter sizes of convolutional layers L_2 and L_3 are $f_2 \times f_2 \times n_1 \times n_2$ and $f_3 \times f_3 \times n_2 \times 1$, respectively. L_3 has only one kernel to obtain the image as output; otherwise, an additional layer with one kernel and other aggregation post-processing would be required.

For the objective function, given a training set composed of high-resolution images I_n^{HR} and their corresponding low-resolution images I_n^{LR}, the pixel mean squared error (MSE) between the reconstructed image and the ground truth is calculated as the training objective:

$$\ell(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{n=1}^{N} \| I_n^{HR} - f(I_n^{LR}; \mathbf{W}, \mathbf{b}) \|^2$$

where $\mathbf{W}$ and $\mathbf{b}$ represent the weights and biases of the video reconstruction model, respectively.

1.2 Pre-training Model

Before training the video reconstruction model, the weights of the reconstruction model are first pre-trained. The image pre-training model has only convolutional layers, with the advantage that the size of input images can be arbitrary and the algorithm is not patch-based, ensuring algorithmic time efficiency.

To correctly initialize the video reconstruction model, assume that instead of using three consecutive video frames, the same frame is used three times as

input. The video reconstruction model and image reconstruction model should produce identical output results. Because L_1 and L_2 are structurally identical to those in the pre-training reconstruction architecture, we only need to ensure that the input data to the convolutional layers is the same for both models. For the pre-training model, the output data from L_1 is represented as h_1. The video reconstruction model calculates the same data as:

$$h_1^v = \sum_{t=-1}^{1} h_1(y_t)$$

By setting $h_1(y_{t-1}) = h_1(y_t) = h_1(y_{t+1})$ in equation (3), if we want equation (1) to equal equation (3), the following condition must be satisfied:

$$\mathbf{W}_1^v = \frac{1}{3}\mathbf{W}_1$$

This is equivalent to performing an averaging operation on the input images before applying convolutional layer L_1.

An ideally motion-compensated frame should be identical to its reference frame. Using this framework to train neural networks would theoretically lead to equal weights for frames y_{t-1}, y_t, and y_{t+1} in L_1 and L_2, particularly when the video frame content contains non-rigid moving objects, which would significantly impact reconstruction quality. If each frame of the video sequence is reconstructed separately, each frame would be positioned at time $t-1$, t, or $t+1$ at a certain moment. Therefore, the motion compensation error from frame y_{t-1} to the current frame y_t should be the same as from frame y_{t+1} to y_t.

This implies that the filters for frame y_{t-1} and frame y_{t+1} in L_1 should have the same weights. Similarly, all filter weights in L_2 should also be identical. Therefore, the weights in L_1 of the model are set to be identical. Applying the same filter settings at specific spatial locations in frame y_t can also represent the same local correlation, essentially extending the convolutional nature of the network in the temporal dimension and enabling filter weight sharing.

The filter sizes of convolutional layer L_1 in the pre-training model and video reconstruction model differ. The first dimension in the video reconstruction model is three times that of the pre-training model because the three input frames are concatenated along the temporal dimension. The kernel width, kernel depth, and number of kernels used in the video reconstruction model are consistent with the pre-training model so that the pre-trained filter values can be directly used in the video reconstruction model. The only difference is that the filters in convolutional layer L_1 of the video reconstruction model use three input frames, so the filter depth in the concatenation layer of the video reconstruction network is three times that of the pre-training model.

Additionally, the output data of convolutional layer L_1 is similar to the output data obtained from single-frame reconstruction because L_2 and L_3 maintain the same settings as in the single-frame reconstruction model.

1.3 Motion Compensation

If large motion shifts or motion blur occur in the video, motion compensation may become difficult, which can also cause boundary effects and artifacts in the high-resolution image reconstruction process, thereby weakening reconstruction quality. The model introduces an adaptive motion compensation method to reduce the influence of adjacent frames during motion estimation. Motion compensation can be performed according to equation (8):

$$y_{amc} = a(m,c) \cdot y_t + (1 - a(m,c)) \cdot y_{mc}$$

where $a(m,c)$ controls the correlation between the center frame and adjacent frames at each pixel position (i,j); y_t is the center frame; y_{mc} is the adjacent frame after motion compensation; and y_{amc} is the video frame after adaptive motion compensation.

$$a(m,c) = \exp(-k \cdot e(m,c))$$

where k is a constant parameter and $e(m,c)$ is the mismatch error of motion compensation. Larger errors may be caused by motion occlusion, object blur, or positions near motion boundaries. According to equations (8) and (9), when the motion compensation error is large at position (i,j), the corresponding weight is small. This means the adaptive motion compensation pixel is primarily the pixel from the current frame, thereby better ensuring the quality of the reconstructed frame.

1.4 Channel Optimization

One method to enhance image resolution is to perform convolution in low-resolution space with a fractional stride of $\frac{1}{r}$, followed by convolution in high-resolution space with a stride of 1. Since convolution operations occur in high-resolution space, the computational overhead of the algorithm increases by r^2 times. Alternatively, convolution can be performed in low-resolution space with a stride of $\frac{1}{r}$ using filters of size $W \times H \times C$, activating different weight parts of the convolution. Weights that fall exactly between pixels are not activated or calculated, and the number of activation patterns is exactly r^2. According to the position of each activation pattern, up to r^2 weights can be activated. Based on different sub-pixel positions (x,y), the convolution operation of the filter on the image periodically activates these patterns, where x, y represent output pixel coordinates in high-resolution space. When $r = 2$, the following method can be used:

$$I^{SR} = f(I^{LR}; \mathbf{W}, \mathbf{b}) = P(S(I^{LR} * W)) + b$$

where P is a periodic operator that rearranges elements of a tensor of dimension $H \times W \times C \cdot r^2$ into a tensor of dimension $Hr \times Wr \times C$. Mathematically, it can be described as:

$$P(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, c \cdot r^2 + \mathrm{mod}(y,r) \cdot r + \mathrm{mod}(x,r)}$$

Therefore, when the convolution operator S has shape $f_2 \times f_2 \times n_1 \times r^2 n_2$, it is equivalent to sub-pixel convolution in low-resolution space with filter W. The output layer directly generates an upscaling filter for each feature map from low-resolution image feature maps, thereby obtaining a high-resolution image.

2 Experiments

The proposed model is first compared with other image and video reconstruction algorithms. Then, the performance of different video reconstruction architectures is quantitatively studied, along with the impact of pre-training and motion compensation mechanisms on video reconstruction quality. Finally, experimental results are compared and analyzed.

2.1 Dataset

The experiments use a publicly available video database containing uncompressed 4K high-resolution video sequence clips (3840×2160 pixels). Fifty-nine scene frames from the videos are selected, with 53 frames used for training and 6 frames for testing. Four frames from each test sequence are used, and the average Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values are calculated across 24 test frames as performance metrics. Experiments sample the video with a scaling factor of 4 to obtain 960×540 pixel resolution images for better comparison with other reconstruction algorithms. An additional video set "Videoset4" is also selected for testing. During experiments, the first and last frame of each video are skipped to ensure that a complete set of 3 consecutive frames is always available as model input during video reconstruction.

2.2 Model Parameter Settings

The model used for pre-training has 3 convolutional layers, with ReLU units following L_1 and L_2. L_1 has 64 kernels with size 9×9; L_2 has 32 kernels with size 5×5; L_3 has one kernel of size 5×5. The filters of the video reconstruction model have the same initial parameter configuration as the image pre-training model.

Images and videos are converted to the YCbCr color space, and only the luminance channel (Y channel) is used for training, testing, and objective performance metric calculation. To establish an effective video training set, 3 sets of consecutive frames are extracted from training video scenes. The required scaling factors of 2, 3, or 4 are implemented in MATLAB for sampling, and the resulting low-resolution frames are upsampled to original resolution using bicubic interpolation. Optical flow from the first and last frames to the center frame is then calculated to obtain motion-compensated frames. From the resulting 3 frames (2 motion-compensated frames and 1 center frame), 36×36×3 data cubes are extracted, representing 36×36 pixel image patches from 3 consecutive frames. If a frame' s patches do not contain sufficient structural information, that frame is removed. The final training database consists of these image data cubes.

To optimize filter weights and biases during training, a loss function to be minimized must be defined. The Euclidean distance between output images from the training dataset and ground truth images needs to be measured, and PSNR performance measurement is also directly related to Euclidean distance. To avoid boundary effects during training, 36×36 pixel patches can be zero-padded, or smaller convolutional outputs can be allowed, meaning each convolutional layer' s output patch shrinks accordingly. The reduced output patches correspond to the central 20×20 pixels of the original patches, and these central pixels are used to calculate the loss function.

In video reconstruction training, experiments use a batch size of 240, with learning rates of 10^{-4} for the first two layers and 10^{-5} for the last layer, and a weight decay rate of $5 \times 10^{-4} \times 10^{-2}$. Using the pre-training mechanism, 5×10^2 iterations are performed, with iterative filter effects for different frames shown in [Figure 3: see original paper]. Experiments show that reducing the learning rate does not further improve reconstruction quality, so the learning rate is kept constant throughout training.

2.3 Comparison with Classical Algorithms

The proposed model is compared with classical single-frame and video reconstruction algorithms, using bicubic interpolation as a baseline. Video super-resolution reconstruction can be achieved by simply applying single-frame reconstruction models to each frame, including Bicubic interpolation, A+ algorithm [9], and SRCNN model [17]. Additionally, comparisons are made with the classical video reconstruction algorithm Bayesian-MB [13], testing all video reconstruction methods using ±1 adjacent frames.

The first implementation variation uses 19×19 pixel bicubic interpolated patches instead of 5×5 pixel input patches, enabling direct comparison with methods using bicubic interpolation input. The second variation replaces the previously used sigmoid activation function with ReLU [23], as the former provides faster model convergence.

[Figure 2: see original paper] shows performance comparisons of different algorithms (PSNR/dB). [Figure 4: see original paper] shows the convergence speed of the proposed model during training compared with other algorithms. As can be seen, the model converges to stable values faster than the SRCNN model, which is also neural network-based. Compared with all other reconstruction algorithms, it can also more efficiently reconstruct images with higher PSNR values. In addition to high training efficiency, during the testing phase, the algorithm is based on feedforward computation of the convolutional neural network, greatly reducing reconstruction efficiency loss caused by numerous iterative operations in traditional methods. With GPU acceleration, the average reconstruction time for a single video frame is approximately 0.26 seconds. Considering both reconstruction accuracy and efficiency, the model can quickly reconstruct video frame images with higher objective evaluation metrics for different video frames, while effectively removing edge and texture blurring in subjective visual quality ([Figure 5: see original paper]).

Tables 1 and 2 record the objective evaluation metrics PSNR and SSIM values for different algorithm models on test videos. As shown, the proposed model achieves higher average metrics in all experiments. On the test dataset, compared with the SRCNN model, improvements are achieved for scaling factors of 2, 3, and 4, particularly for high-resolution reconstruction. On the Videoset4 dataset, the gains for scaling factors of 2 and 3 are more pronounced than for scaling factor of 4. Experimental results indicate that the improvement in video reconstruction quality is more significant for low scaling factors, possibly due to more accurate motion compensation algorithms in the model. Figures 2 and 5 show examples of reconstructed video test sets using different algorithms and their corresponding PSNR values. Compared with other algorithms, the proposed reconstruction model can better recover more image detail content, clearly removing blur effects at image edges and further enhancing detail clarity, providing better support for subsequent image processing tasks.

3 Conclusion

This paper fully utilizes spatial and temporal information among multiple video frames and proposes a novel video reconstruction model based on convolutional neural networks. By comparing different model structures and employing motion compensation processing and pre-training strategies for model input frames, video reconstruction quality is effectively improved and training time is reduced. For fast-moving objects in videos, an adaptive motion compensation scheme is introduced to handle motion blur caused by such movements. Experiments demonstrate that compared with classical methods, the proposed model achieves superior objective evaluation metrics and higher image quality in video image reconstruction, providing a foundation for other visual tasks and application scenarios requiring high-resolution images. Further improvement and optimization of the model, particularly for super-resolution reconstruction and visual applications on video data from specific application scenarios, represent future

research directions.

References

[1] He Xiaohai, Wu Yuanyuan, Chen Weilong, et al. Overview of video super-resolution reconstruction technology [J]. Information and Electronic Engineering, 2011, 9(1): 1-6.

[2] Belekos S P, Galatsanos N P, Katsaggelos A K. Maximum a posteriori video super-resolution using a new multichannel image prior [J]. IEEE Trans on Image Process, 2010, 19(6): 1451-1464.

[3] Babacan S D, Molina R, Katsaggelos A K. Variational Bayesian super resolution [J]. IEEE Trans on Image Process, 2011, 20(4): 984-999.

[4] Liu Ce, Sun Deqing. On Bayesian adaptive video super resolution [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2014, 36(2): 346-360.

[5] Yang J, Wright J, Huang T, et al. Image super-resolution via sparse representation [J]. IEEE Trans on Image Process, 2010, 19(11): 2861-2873.

[6] Aharon M, Elad M, Bruckstein A. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation [J]. IEEE Trans on Signal Process, 2006, 54(11): 4311-4322.

[7] Song B C, Jeong S C, Choi Y. Video super-resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training [J]. IEEE Trans on Circuits and Systems for Video Technology, 2011, 21(3): 274-285.

[8] Zhang Yan, Li Jianzeng, Li Deliang, et al. UAV reconnaissance video super-resolution reconstruction method [J]. Journal of Image and Graphics, 2016, 21(7): 967-976.

[9] Timofte R, De Smet V, Van Gool L. A+: adjusted anchored neighborhood regression for fast super-resolution [C]// Proc of the 12th IEEE Asian Conference on Computer Vision. 2014: 1920-1927.

[10] Schulter S, Leistner C, Bischof H. Fast and accurate image upscaling with super-resolution forests [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3791-3799.

[11] Glasner D, Bagon S, Irani M. Super-resolution from a single image [C]// Proc of IEEE International Conference on Computer Vision. 2009: 349-356.

[12] Qin Fengqing, He Xiaohai, Chen Weilong, et al. A video super-resolution reconstruction method based on sub-pixel registration [J]. Journal of Optoelectronics · Laser, 2009, 20(7): 972-976.

[13] Ma Z, Jia J, Wu E. Handling motion blur in multi-frame super-resolution [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2015: 5224-5232.

[14] Takeda H, Milanfar P, Protter M, et al. Super-resolution without explicit subpixel motion estimation [J]. IEEE Trans on Image Process, 2009, 18(9): 1451-1464.

[15] Shi W, Caballero J, Huszar F, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1874-1883.

[16] Kappeler A, Yoo S, Dai Q, et al. Video super-resolution with convolutional neural networks [J]. IEEE Trans on Computational Imaging, 2016, 2: 109-122.

[17] Dong Chao, Loy C C, He Kaiming, et al. Learning a deep convolutional network for image super-resolution [C]// Proc of IEEE European Conference on Computer Vision. 2014: 184-199.

[18] Wang Zhangyang, Yang Yingzhen, Wang Zhaowen, et al. Self-tuned deep super resolution [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-8.

[19] Cui Z, Chang H, Shan S, et al. Deep network cascade for image super-resolution [C]// Proc of IEEE European Conference on Computer Vision. 2014: 1-16.

[20] Cheng Minghui, Lin Naiwei, Hwang K, et al. Fast video super-resolution using artificial neural networks [C]// Proc of the 8th International Symposium on Communication Systems, Networks & Digital Signal Processing. 2012: 1-4.

[21] Liao R, Tao X, Li R, et al. Video super-resolution via deep draft-ensemble learning [C]// Proc of IEEE International Conference on Computer Vision. 2015: 531-539.

[22] Huang Xuan, Yang Xiaomei. Video super-resolution reconstruction based on low rank and total variation [J]. Application Research of Computers, 2015, 32(3): 938-941.

[23] Maas A, Hannun A, Ng A. Rectifier nonlinearities improve neural network acoustic models [C]// Proc of IEEE International Conference on Machine Learning. 2013: 1-8.

Note: Figure translations are in progress. See original paper for figures.