Spatial-Temporal Recurrent Residual Networks for Video Super-Resolution

Wenhan Yang, Jiaying Liu, and Zongming Guo**

Peking University, Beijing, P.R.China, 100871

Abstract. In this paper, we propose a new video Super-Resolution (S-R) method by jointly modeling intra-frame redundancy and inter-frame motion context in a unified deep network. Different from conventional methods, the proposed Spatial-Temporal Recurrent Residual Network (STR-ResNet) investigates both spatial and temporal residues, which are represented by the difference between a high resolution (HR) frame and its corresponding low resolution (LR) frame and the difference between adjacent HR frames, respectively. This spatial-temporal residual learning model is then utilized to connect the intra-frame and inter-frame redundancies within video sequences in a recurrent convolutional network and to predict HR temporal residues in the penultimate layer as guidance to benefit estimating the spatial residue for video SR. Extensive experiments have demonstrated that the proposed STR-ResNet is able to efficiently reconstruct videos with diversified contents and complex motions, which outperforms the existing video SR approaches and offers new state-of-the-art performances on benchmark datasets.

Keywords: Spatial residue, temporal residue, video super-resolution, inter-frame motion context, intra-frame redundancy

1 Introduction

Video super-resolution (SR) aims to produce high-resolution (HR) video frames from a sequence of low-resolution (LR) inputs. It is modeled as restoring the original scene $\mathbf{x}_t$ from its several quality-degraded observations $\{\mathbf{y}_t\}$. Typically, the observation can be modeled as

$$\mathbf{y}_t = \mathbf{D}_t\mathbf{x}_t + \mathbf{v}_t, t = 1, \ldots, T. \tag{1}$$

Here $\mathbf{D}_t$ encapsulates various signal quality degradation factors at the time instance t, *e.g.*, motion blur, defocus blur and down-sampling. Additive noise during observation at that time is denoted as $\mathbf{v}_t$. Generally, the SR problem, *i.e.*, solving out $\mathbf{x}_t$ in Eq. (1), is an ill-posed linear inverse problem that is rather challenging. Thus, accurately estimating $\mathbf{x}_t$ demands either sufficient observations $\mathbf{y}_t$ or proper priors on $\mathbf{x}_t$.

** Corresponding author. This work was supported by National Natural Science Foundation of China under contract No. U1636206.

All video SR methods can be divided into two classes: reconstruction-based and learning-based. Reconstructed-based methods [1, 4, 5] craft a video SR process to solve the inverse estimation problem of (1). They usually perform motion compensation at first, then perform deblurring by estimating blur functions in $\mathbf{D}_t$ of (1), and finally recover details by local correspondences. The hand-crafted video SR process cannot be applicable for every practical scenario of different properties and perform not well to some unexpected cases.

In contrast, learning-based methods handle the ill-posed inverse estimation by learning useful priors for video SR from a large collection of videos. Typical methods include recently developed deep learning-based video SR methods [8–10] and give some examples of non-deep learning approaches. In [8], a funnel shape convolutional neural network (CNN) was developed to predict HR frames from LR frames that are aligned by optical flow in advance. It shows superior performance on recovering HR video frames captured in still scenes. However, this CNN model suffers from high computational cost (as it relies on time-consuming regularized optical flow methods) as well as visual artifacts caused by complex motions in the video frames. In [9,10], a bidirectional recurrent convolutional network (BRCN) was employed to model the temporal correlation among multiple frames and further boost the performance for video SR over previous methods.

However, previous learning-based video SR methods that learn to predict HR frames directly based on LR frames, suffer from following limitations. First, these methods concentrate on exploiting between-frame correlations and does not *jointly* consider the intra- and inter-frame correlations that are both critical for the quality of video SR. This unfavorably limits the capacity of the network for recovering HR frames with complex contents. Second, the successive input LR frames are usually highly correlated with the whole signal of the HR frames, but are not correlated with the high frequency details of these HR images. In the case where dominant training frames present slow motion, the learned priors hardly capture hard cases, such as large movements and shot changes, where neighboring frames distinguished-contributed operations are needed. Third, it is desirable for the joint estimation of video SR to impose priors on missing high frequency signals. However, in previous methods, the potential constraints are directly enforced on the estimated HR frames.

To solve the above-mentioned issues, in this work, we propose a unified deep neural network architecture, **S**patial **T**emporal **R**ecurrent **Res**idual **Net**work (STR-ResNet), to *jointly* model the intra-frame and the inter-frame correlation in an end-to-end trainable manner. Compared with previous (deep) video SR methods [8–10], our proposed deep network model does not require explicit computation of optical flow or motion compensation. In addition, our proposed model unifies the convolutional neural networks (CNNs) and recurrent neural networks (RNNs) which are known to be powerful in modeling sequential data. Combining the spatial convolutional and temporal recurrent architectures enables our model to capture spatial and temporal correlations jointly. Specially, it models spatial and temporal correlations among multiple video frames jointly. The temporal residues of HR frames are predicted based on input LR frames

along with their temporal residues to further regularize estimation of the spatial residues. This architectural choice enables the network to handle the videos containing complex motions in a moving scene, offering pleasant video SR results with few artifacts in a time-efficient way. Extensive experiments on video SR benchmark datasets clearly demonstrate the contribution of each component to the overall performance.

2 Spatial-Temporal Recurrent Residual Networks for Multi-Frame SR

In this section, a basic network structure – SRes-CNN for spatial residual learning for single image SR is presented in formulation. Then, we construct a new proposed STR-ResNet by stacking and connecting the basic component – SRes-CNN for joint temporal learning is elaborated.

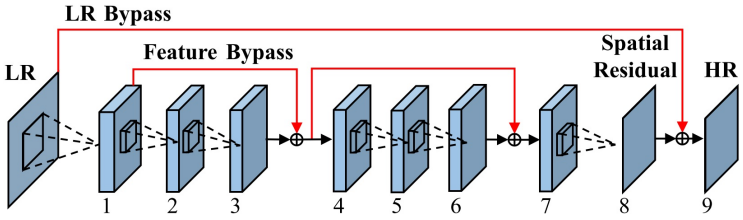


Fig. 1: The bypass structure and spatial residual learning in the proposed SRes-CNN. The *feature bypass* connection forwards the feature maps output from a previous layer (1st / 4th) to a later one (4th / 7th). The *LR bypass* from the LR frame to the last layer (9th) makes the network focus on predicting the residue, the high frequency component of a frame.

2.1 Architecture of SRes-CNN

Single frame SR aims to reconstruct an HR frame from a single LR frame. Some recent deep learning based SR methods [13–15] propose to use a CNN model to extract features from LR frames and then map them to HR ones. In our paper, we propose a new CNN architecture – Spatial Residual CNN (SRes-CNN) – to learn spatial residue between HR and LR frames. Specifically, SRes-CNN contains nice layers, including six convolutional layers, three bypass connections and three element-wise summations, as shown in Fig. 1. The bypass connections forward the feature maps output from the i-th layer ($i = 1, 4$ for the SRes-CNN we use in the experiments) to the $(i + 2)$-th layer directly. Then, the feature maps output from the $(i + 2)$-th and i-th layers are fused as input to the next $(i + 3)$-th convolution layer. To focus on predicting the high-frequency components, SRes-CNN also establishes a bypass connection from the input LR frame to the penultimate layer. Note that, these two kinds of bypass connections play different roles in STR-ResNet. The first "long-range"one directly forwards an input LR frame to its penultimate layer (the 7th one). The other bypass connections provide a coarse-to-fine refinement. For example, the feature maps of

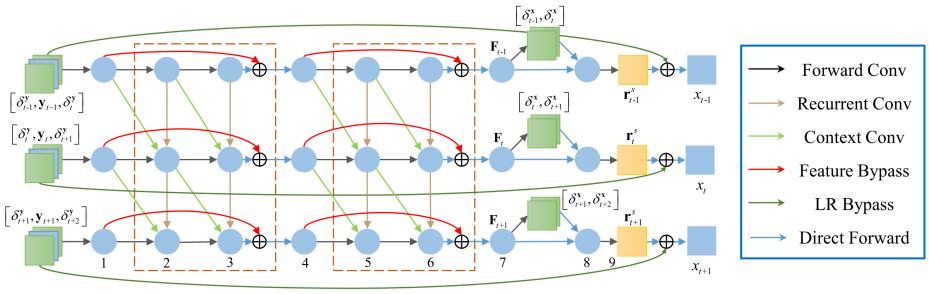


Fig. 2: The architecture of the STR-ResNet. It has a two-layer structure, which includes spatial and temporal residuals jointly in a unified deep framework. (Best viewed in color.)

the 1st layer correspond to the low-level features directly extracted from the LR image, and then the feature maps of the 3rd and 5th layers therefore concentrate on capturing the enhanced details of HR features. Besides, the bypass connections also make constructing a deeper network possible and speed up the training process [13].

2.2 Architecture of STR-ResNet

We now elaborate how the STR-ResNet exploits inter-frame correlation by connecting multiple SRes-CNNs with convolutions and how it incorporates temporal residual information for multi-frame SR. The intuition of choosing the architecture is to propagate information across multiple frames recurrently in order to capture the temporal context. STR-ResNet uses recurrent units to connect several SRes-CNNs to embed the temporal correlation. The STR-ResNet takes not only the LR frames but also the differences of adjacent LR frames as inputs. It reconstructs an HR frame through fusing its corresponding LR frame and the predicted spatial residue, under the guidance of the predicted temporal residues among adjacent frames. As shown in Fig. 2, STR-ResNet performs following six types of operations:

1. **Forward convolution**. The convolution operations in each SRes-CNN component for single frame SR.
2. **Recurrent convolution**. To propagate information across adjacent frames and restore lost information from the adjacent frames, STR-ResNet performs recurrent convolutions (the gray arrows between frames as shown in Fig. 2) to propagate the features of the i-th layer of the adjacent $(t-1)$-st and $(t+1)$-st frames (defined as $\mathbf{C}^{a}_{(t-1,i)}$ and $\mathbf{C}^{a}_{(t+1,i)}$) to the i-th layer of the t-th frame (defined as $\mathbf{C}^{r,p}_{(t,i)}$ and $\mathbf{C}^{r,n}_{(t,i)}$.)
3. **Context convolution**. With the similar intuition of transmitting complementary information among frames, the context convolution (the light-green arrows between frames as shown in Fig. 2) propagates the features of the $(i-1)$-th layer of the adjacent $(t-1)$-st and $(t+1)$-st frames (defined as $\mathbf{C}^{a}_{(t-1,i-1)}$ and $\mathbf{C}^{a}_{(t+1,i-1)}$) to the i-th layer of the t-th frame (defined as $\mathbf{C}^{c,p}_{(t,i)}$ and $\mathbf{C}^{c,n}_{(t,i)}$.)

4. **Temporal residue embedding**. In the 8th layer, we first predict the temporal residues (the green rectangles between the 7-th and 8-th layers as shown in Fig. 2). In the training, these outputs are constrained by the loss function to regress the ground-truth temporal residues, which will be presented more clearly in the next subsection. Then, we concatenate the predicted temporal residues with the output feature maps from the 7th layer to generate the output feature maps of the 8th layer.
5. **Feature bypass**. The operation to transmit the features output from the 1st/4th layers and combine them with the output of the 3rd/6th layers respectively.
6. **LR bypass**. It bypasses the LR frames to the output of the 8th layer, which generates the estimated HR details of frame t.
7. **Feed forward**. The operation to propagate the feature maps to the subsequent unit.

Among these operations, the recurrent and context convolutions are only deployed in the 2nd, 3rd, 5th and 6th layers of SRes-CNNs as shown in Fig. 2 (b). All the recurrent connections transmit outputs of layers (2nd, 3rd, 5th and 6th) on the t-th frame to their corresponding layers (2nd, 3rd, 5th and 6th) of the adjacent $(t-1)$-th and $(t+1)$-th frames. All the context connections transmit from a previous layer (1st, 2nd, 4th and 5th) of the t-th frame to their corresponding next layer (2nd, 3rd, 5th and 6th) of the adjacent $(t-1)$-th and $(t+1)$-th frames. After all these convolutions, an element-wise summation operation is employed to fuse these convolution outputs and produce a new feature map. The outputs of the five convolutional operations and the fusion are formulated as follows,

$$\begin{aligned}
\mathbf{C}^{f}_{(t,i)} &= \mathbf{W}^{f}_{i} * \mathbf{C}^{a}_{(t,i-1)} + \mathbf{B}^{f}_{i}, \\
\mathbf{C}^{c,p}_{(t,i)} &= \mathbf{W}^{c,p}_{i} * \mathbf{C}^{a}_{(t-1,i-1)} + \mathbf{B}^{c,p}_{i}, \\
\mathbf{C}^{c,n}_{(t,i)} &= \mathbf{W}^{c,n}_{i} * \mathbf{C}^{a}_{(t+1,i-1)} + \mathbf{B}^{c,n}_{i}, \\
\mathbf{C}^{r,p}_{(t,i)} &= \mathbf{W}^{r,p}_{i} * \mathbf{C}^{a}_{(t-1,i)} + \mathbf{B}^{r,p}_{i}, \\
\mathbf{C}^{r,n}_{(t,i)} &= \mathbf{W}^{r,n}_{i} * \mathbf{C}^{a}_{(t+1,i)} + \mathbf{B}^{r,n}_{i}, \\
\mathbf{C}^{a}_{(t,i)} &= \max\left(0, \mathbf{C}^{f}_{(t,i)} + \mathbf{C}^{c,p}_{(t,i)} + \mathbf{C}^{c,n}_{(t,i)} + \mathbf{C}^{r,p}_{(t,i)} + \mathbf{C}^{r,n}_{(t,i)}\right),
\end{aligned} \tag{2}$$

where $i = 2, 3, ..., 6$, and $\mathbf{W}$ and $\mathbf{B}$ are filters and biases, respectively. The superscripts f, c, r and a denote the unit type – forward convolution, context convolution, recurrent convolution and element-wise summation aggregation. The superscripts p, n denote the direction of the convolution, from the previous frame or the next frame. The subscript (t, i) denotes that the operation is performed on the i-th layer of the t-th frame. Consequently, $\mathbf{C}^{f}_{(t,i)}$, $\mathbf{C}^{c,p}_{(t,i)}$, $\mathbf{C}^{c,n}_{(t,i)}$, $\mathbf{C}^{r,p}_{(t,i)}$ and $\mathbf{C}^{r,n}_{(t,i)}$ are the outputs of the forward convolution, context convolution from the previous frame, context convolution from the next frame, recurrent convolution from the previous frame and recurrent convolution from the next frame in the i-th layer of the t-th frame respectively. $\mathbf{C}^{a}_{(t,i)}$ performs an element-wise summation overall all the five outputs, for combining the predictions from the current

frame and adjacent frames. A ReLU unit is then connected subsequently. The responses of previous layers are as follows,

$$\mathbf{C}^a_{(t,i)} = \mathbf{C}^f_{(t,i)}, \text{ for } i = 1, 4, 7, 9. \tag{3}$$

For the 8st layer, we try to predict the temporal residues of HR frames and utilize them as parts of the features to estimate the spatial residues,

$$\delta^{\mathbf{x}}_t = \mathbf{W}_\delta * \mathbf{C}^a_{(t,7)} + \mathbf{b}_\delta, \mathbf{C}^a_{(t,8)} = \left[\mathbf{C}^a_{(t,7)}, \delta^{\mathbf{x}}_t\right]. \tag{4}$$

With the help of context and recurrent convolutions as well as the temporal residue constraints, the STR-ResNet captures the inter-frame motion context propagated from adjacent frames for video SR.

2.3 Training STR-ResNet

To learn meaningful features and capture some consistent motion contexts between frames, STR-ResNet shares its parameters among different frames. That is, for all $\mathbf{C}^f_{(t,i)}, \mathbf{C}^{c,p}_{(t,i)}, \mathbf{C}^{c,n}_{(t,i)}, \mathbf{C}^{r,p}_{(t,i)}$, and $\mathbf{C}^{r,n}_{(t,i)}$, their parameters $\left\{\mathbf{W}^f_i, \mathbf{B}^f_i\right\}$, $\{\mathbf{W}^{c,p}_i, \mathbf{B}^{c,p}_i\}$, $\{\mathbf{W}^{c,n}_i, \mathbf{B}^{c,n}_i\}$, $\{\mathbf{W}^{r,p}_i, \mathbf{B}^{r,p}_i\}$ and $\{\mathbf{W}^{r,n}_i, \mathbf{B}^{r,n}_i\}$, are decided by the unit type, denoted by superscript, and layer depth, and have nothing to do with the frame number.

For training STR-ResNet, provided with LR video frames $\{\mathbf{y}^g_t\}$ and HR frames $\{\mathbf{x}^g_t\}$, we minimize the Mean Square Error (MSE) between the predicted frames and the ground truth HR frames:

$$\min_{\mathbf{\Theta}} \sum_{t=1}^{9} \lambda_t \|\widehat{\mathbf{x}}_t\left(\mathbf{y}^g_t, \mathbf{\Theta}\right) + \mathbf{y}^g_t - \mathbf{x}^g_t\|^2_F + c \sum_{t=1}^{9} \|\widehat{\delta}^{\mathbf{x}}_t\left(\mathbf{y}^g_t, \mathbf{\Theta}\right) + \mathbf{x}^g_t - \mathbf{x}^g_{t-1}\|^2_F, \tag{5}$$

where

$$\mathbf{\Theta} = \left(\mathbf{W}^f, \mathbf{B}^f, \mathbf{W}^{c,p}, \mathbf{B}^{c,p}, \mathbf{W}^{c,n}, \mathbf{B}^{c,n}, \mathbf{W}^{r,p}, \mathbf{B}^{r,p}, \mathbf{W}^{r,n}, \mathbf{B}^{r,n}\right), \tag{6}$$

$\mathbf{x}^g_0 = \mathbf{x}^g_1$ and $\{\lambda_i, i = 1, 2, ..., 8, 9\}$ are the weighting parameters that control the relative importance of these terms. c is set to 0.1 to play a role but not the dominant one. We set $n_T = 9$ as the step/recurrence number following the default setting in the previous RNN-based video SR method [9].

3 Experiments

3.1 Experiments Setting

The compared single image SR baselines include Bicubic interpolation and super-resolution convolution neural network (SRCNN) [13]. The compared video SR baselines include a commercial software video enhancer (VE)[1], 3DSKR [19],

[1] http://www.infognition.com/videoenhancer/

Draft SR [8] and BRCN [9]. To evaluate the effectiveness of our method, we simulate the degradation process and enlarge the generated LR images to their original scales. Peak Signal to Noise Ratio (PSNR) is chosen as the metric. The testing scaling factor is chosen as 4. In the simulation of degradation, the LR frames are generated by blurring HR frames with a 9×9 Gaussian filters with blur level 1.6.

For training our STR-ResNet, we use 300 collected video sequences, sampled uniformly from 30 high-quality 1080p HD video clips as our training set[2][3]. We use 6 HDTV sequences (*Tractor*, *Sunflower*, *Blue Sky*, *Station*, *Pedestrian*, *Rush Hour*) downloaded from the Xiph.org Video Test Media[2] as the testing set, which are commonly used high quality video sequences for video coding testing. To reduce the memory storage needed in the training phrase, we crop these frame groups into 75,000 overlapped patch groups as the input of training. Each patch group contains 9 adjacent patches in the temporal domain with the same location in the spatial domain. Similar to [13], the size of the spatial window of each patch group is set to 33×33 and the spatial stride is set to 11.

The proposed STR-ResNet uses the following parameters: all convolutions have a kernel size of 3×3 and a padding size 1; the layer type and number are set as mentioned above; the channel size of the intermediate layers is set to 64. We employ stochastic gradient descent [4] to train the whole network. The training strategy is standard: learning rates of weights and biases of these filters are set to 0.0001 initially and decrease to 0.00001 after 2.5×10^5 iterations (about 37 epochs). We stop the training in 3×10^5 iterations (about 44 epochs). The batch size is set to 6.

3.2 Objective Evaluation

Tables 1 shows PSNR results of compared video super-resolution methods on the testing image set. The proposed method and the BRCN method are evaluated with 9 adjacent frames as inputs. For Draft Learn, we report its results in two cases: 1) taking 31 adjacent LR frames (Draft31) as its input; 2) taking 5 adjacent LR frames (Draft5) as its input. For 3DSKR, the HR estimation is generated based on adjacent 5 LR frames. From the result, one can observe that even compared with the recent Draft Learning and BRCN, our proposed STR-ResNet achieves a significant performance gain over them. In particular, the average gain over the second best BRCN is as high as 0.56dB. VE and 3DKR achieve better reconstructed results than Bicubic. However, their PSNR results are lower than very recent single image SR methods, such as SRCNN and A+, which only make use of the intra-frame spatial correlation.

[2] **Xiph.org Video Test Media [derf's collection]**, https://media.xiph.org/video/derf/

[3] **Dataset from Harmonic Inc.**, http://www.harmonicinc.com/resources/videos/4k-video-clip-center

[4] http://caffe.berkeleyvision.org/tutorial/solver.html

Table 1: PSNR results among different methods for Video SR (scaling factor: 4). The bold numbers denote the best performance.

Video	Bicubic	SRCNN	VE	3DSKR	Draft	BRCN	STR-ResNet
Tractor	31.10	32.13	31.27	32.27	30.34	33.23	**33.85**
Sunflower	37.85	38.69	37.55	37.57	36.43	39.28	**40.02**
Blue Sky	28.77	30.16	29.19	29.74	30.92	31.40	**32.23**
Station	33.35	34.38	33.36	34.80	33.22	35.20	**35.63**
Pedestrian	33.51	34.55	33.60	33.91	31.78	34.95	**35.22**
Rush Hour	38.17	38.90	37.96	37.49	36.22	39.86	**40.30**
Average	33.79	34.80	33.82	34.30	33.15	35.65	**36.21**

3.3 Subjective Evaluation

Figs. 3 visualize the SR results of different methods. Bicubic generates blurred results. SRCNN generate sharper results. However, without exploiting the temporal correlation, some visually important features are blurred, such as the brand text in Fig. 3. In contrast, video SR methods, such as 3DSKR and Draft Learning, generate results with richer details. But 3DSKR may suffer from inaccurate motion estimation and generate block artifacts, and Draft Learning produces granular artifacts in smooth regions, where optical flow estimation is unreliable. Due to RNN's strong capacity of modeling complex motions, BRCN and our method present rather sharp results. Especially, the proposed STR-ResNet recovers details with a very natural look.

4 Conclusion and Future Work

In this paper, we proposed a novel Spatial-Temporal Recurrent Residual Network (STR-ResNet) for video super-resolution. This network simultaneously models high frequency details of single frames, the differences between high resolution (HR) and low resolution (LR) frames, as well as the changes of these adjacent detail frsames. To model intra-frame correlation, a CNN structure with bypass connections is constructed to learn spatial residual of a single frame. To model inter-frame correlation, STR-ResNet estimates the temporal residue implicitly. Extensive experiments have demonstrated the effectiveness and efficiency of our method for video SR.

References

1. C. Liu, D. Sun, On bayesian adaptive video super resolution, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2) (2014) 346–360.
2. S. Baker, T. Kanade, Super-Resolution Optical Flow (1999).
3. H. He, L. P. Kondi, An image super-resolution algorithm for different error levels per frame, IEEE Transactions on Image Processing 15 (3) (2006) 592–603.
4. A. V. Kanaev, C. W. Miller, Multi-frame super-resolution algorithm for complex motion patterns, Opt. Express 21 (17) (2013) 19850–19866.
5. O. A. Omer, T. Tanaka, Region-based weighted-norm approach to video super-resolution with adaptive regularization, in: Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing, 2009, pp. 833–836.

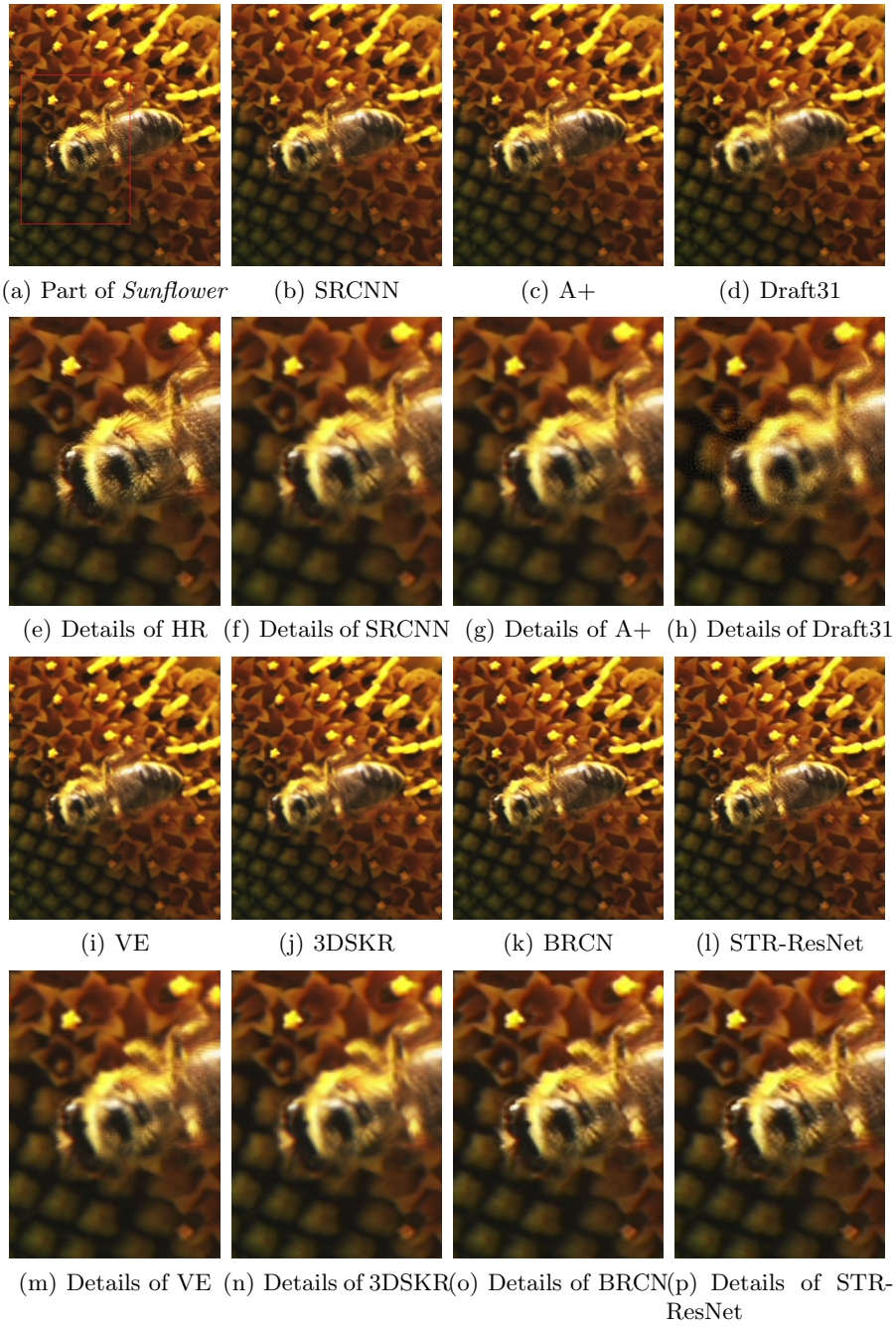
(a) Part of *Pedestrian* (b) SRCNN (c) A+ (d) Draft5

(e) Details of HR (f) Details of SRCNN (g) Details of A+ (h) Details of Draft5

(i) VE (j) 3DSKR (k) BRCN (l) STR-ResNet

(m) Details of VE (n) Details of 3DSKR (o) Details of BRCN (p) Details of STR-ResNet

Fig. 3: The reconstruction results of *Pedestrian* with different methods (enlarge factor: 4×).



(a) Part of *Sunflower* (b) SRCNN (c) A+ (d) Draft31

(e) Details of HR (f) Details of SRCNN (g) Details of A+ (h) Details of Draft31

(i) VE (j) 3DSKR (k) BRCN (l) STR-ResNet

(m) Details of VE (n) Details of 3DSKR (o) Details of BRCN (p) Details of STR-ResNet

Fig. 4: The reconstruction results of *Sunflower* with different methods (enlarge factor: 4×).

(a) Part of *Tractor* (b) SRCNN (c) A+ (d) Draft31

(e) Details of HR (f) Details of SRCNN (g) Details of A+ (h) Details of Draft31

(i) VE (j) 3DSKR (k) BRCN (l) STR-ResNet

(m) Details of VE (n) Details of 3DSKR (o) Details of BRCN (p) Details of STR-ResNet

Fig. 5: The reconstruction results of *Tractor* with different methods (enlarge factor: 4×).

6. S. Farsiu, M. D. Robinson, M. Elad, P. Milanfar, Fast and robust multiframe super resolution, IEEE Transactions on Image Processing 13 (10) (2004) 1327–1344.
7. L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: Nonlinear Phenomena 60 (14) (1992) 259 – 268.
8. R. Liao, X. Tao, R. Li, Z. Ma, J. Jia, Video super-resolution via deep draft-ensemble learning, in: Proc. IEEE Int'l Conf. Computer Vision, 2015, pp. 531–539.
9. Y. Huang, W. Wang, L. Wang, Bidirectional recurrent convolutional networks for multi-frame super-resolution, in: Proc. Annual Conference on Neural Information Processing Systems, 2015, pp. 235–243.
10. Y. Huang, W. Wang, L. Wang, Video super-resolution via bidirectional recurrent convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (99) (2017) 1–1.
11. W. Wu, C. Zheng, Single image super-resolution using self-similarity and generalized nonlocal mean, in: IEEE International Conference of IEEE Region, 2013, pp. 1–4.
12. W. Dong, L. Zhang, G. Shi, Centralized sparse representation for image restoration, in: Proc. IEEE Int'l Conf. Computer Vision, 2013, pp. 1259–1266.
13. C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, in: Proc. IEEE European Conf. Computer Vision, 2014.
14. Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proc. IEEE Int'l Conf. Computer Vision, 2015, pp. 370–378.
15. W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, S. Yan, Deep edge guided recurrent residual learning for image super-resolution, arXiv:1604.08671.
16. J. Kim, J. K. Lee, K. M. Lee, Deeply-recursive convolutional network for image super-resolution, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2016, pp. 1637–1645.
17. R. Liao, X. Tao, R. Li, Z. Ma, J. Jia, Video super-resolution via deep draft-ensemble learning, in: Proc. IEEE Int'l Conf. Computer Vision, 2015, pp. 531–539.
18. R. Timofte, V. De Smet, L. Van Gool, A+: Adjusted anchored neighborhood regression for fast super-resolution, in: Proc. IEEE Asia Conf. Computer Vision, 2014.
19. H. Takeda, P. Milanfar, M. Protter, M. Elad, Super-resolution without explicit subpixel motion estimation, IEEE Transactions on Image Processing 18 (9) (2009) 1958–1975.
20. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM international conference on Multimedia, 2014, pp. 675–678.
21. Image super-resolution using deep convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2) (2016) 295–307.
22. J. Kim, J. K. Lee, K. M. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2016, pp. 1646–1654.
23. Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proc. IEEE Int'l Conf. Computer Vision, 2015, pp. 370–378.
24. W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution, ArXiv e-prints.
25. R. Timofte, V. D. Smet, L. V. Gool, Semantic super-resolution: When and where is it useful?, Computer Vision and Image Understanding 142 (2016) 1 – 12.