Video Super-Resolution via Deep Draft-Ensemble Learning

Renjie Liao[†] Xin Tao[†] Ruiyu Li[†] Ziyang Ma[§] Jiaya Jia[†]

[†] The Chinese University of Hong Kong
[§] University of Chinese Academy of Sciences
http://www.cse.cuhk.edu.hk/leojia/projects/DeepSR/

Abstract

We propose a new direction for fast video super-resolution (VideoSR) via a SR draft ensemble, which is defined as the set of high-resolution patch candidates before final image deconvolution. Our method contains two main components – i.e., SR draft ensemble generation and its optimal reconstruction. The first component is to renovate traditional feedforward reconstruction pipeline and greatly enhance its ability to compute different super-resolution results considering large motion variation and possible errors arising in this process. Then we combine SR drafts through the nonlinear process in a deep convolutional neural network (CNN). We analyze why this framework is proposed and explain its unique advantages compared to previous iterative methods to update different modules in passes. Promising experimental results are shown on natural video sequences.

1. Introduction

Video super-resolution (VideoSR), a.k.a. multi-frame super-resolution (MFSR), refers to the process of estimating a high resolution (HR) image from a sequence of low resolution (LR) observations. It is fundamental in visual processing, as several applications, including video enhancement and text/object recognition in surveillance and phone videos, can benefit from it. Although effective strategies have been proposed, VideoSR remains a difficult problem for real-world natural image sequences.

Previous methods [17, 7, 18, 5, 6, 15, 14, 16, 12, 13] in this area involve a few key components for pixel-level motion estimation on LR images, warping each LR image to the HR space, and final deconvolution given the physical LR generation process from HR images. They are similarly important since any of them being the bottleneck could degrade system performance and lower the result quality.

Difficulties It has been observed for long time that motion estimation critically affects MFSR. Erroneous motion inevitably distorts local structure and misleads final reconstruction. Albeit essential, in natural video sequences, sufficiently high quality motion estimation is not easy to obtain even with state-of-the-art optical flow methods. On the other hand, the reconstruction and point spread function (PSF) kernel estimation steps could introduce visual artifacts given any errors produced in this process or the low-resolution information is not enough locally.

These difficulties make methods finely modeling all constraints in *e.g.*, [12], *generatively* involve the process with sparse regularization for iteratively updating variables – each step needs to solve a set of nonlinear functions. The computation cost is high.

Our Non-iterative Framework Our idea is to reduce the majority of the computation by employing a non-iterative procedure. It is essentially different from other solutions for VideoSR.

Our principled strategy is to decompose the overall procedure of VideoSR into only two components – unlike conventional methods – based on an ensemble of *SR drafts* in generation and discrimination respectively. SR draft is defined as a HR image patch candidate before the final deconvolution. Our first component in the framework thus is *draft-ensemble* generation, which quickly produces a set of SR drafts from the input LR sequence.

The second component, which is similarly important, is to determine the optimal portfolio among the proposed SR drafts. We propose a deep convolutional neural network (CNN) for its learning. This SR draft-ensemble CNN also integrates the function of deconvolution to form the final HR image with minimal visual artifacts.

Our SR draft-ensemble CNN considers contextual information provided from external data for super-resolution. Note in several previous methods, optimal states have to be reached via iterative processing based on the image generation principle. Our method is along another line to pro-

pose sufficiently many draft proposals in generation and then find the optimal ones via deep learning, which utilizes the strength of discriminative learning.

We experiment with many real-world natural video sequences to validate our new framework. Visually compelling results with many structural details are produced quickly. Moreover, it avoids parameter tuning in the test phase and performs consistently well on both synthetic and real-word data.

2. Related Work

We discuss the relationship and inherent difference between our SR draft ensemble method and existing representative reconstruction-based systems. Our method is related to the fast feedforward maximum likelihood estimation [5], 3D kernel regression (3DKR) [16], and Bayesian adaptive video super-resolution [12].

Our image generation pipeline is based on the feedforward maximum likelihood solution [5]. But rather than relying on possibly erroneous motion estimation, our CNN finds the best among multiple local HR structures provided by the SR draft ensemble. This scheme also extends the capability from handling pure translation [5] to complex local motion. Besides, deconvolution is integrated in the unified network rather than separate employment [9, 18, 5]. Thus during training, our method enlists the ability to adapt deconvolution since all parameters are learned consistently within the same optimization framework. Errors possibly arising in separate steps can be largely reduced.

The 3DKR method [16] upsamples and roughly warps the LR sequence and then adopts local data-driven 3D kernel regression to estimate the HR image. Pixel-level correspondence is still necessary. The 3D kernel is only determined by pixels in the local neighborhood, while our network is learned from a large amount of external data.

As for the Bayesian adaptive method [12], it iteratively estimates all operators in the maximum a posterior (MAP) manner. It provides considerable HR details when optical flow estimate is accurate. To this end, parameters may need to be tuned for the best performance. Differently, our method has two parts for sufficient HR candidate generation and discriminative learning to naturally find suitable estimates. It thus does not need complicated nonlinear optimization during testing and runs much faster. The result quality is also high even for the challenging natural data.

3. SR Draft-Ensemble and Its Analysis

Our method contains two major components for SR draft-ensemble generation and its discrimination. We describe the first component in this section along with its statistical analysis.

Algorithm 1 : SR Draft Generation

1: **For all** i **except the reference frame**
2: Compute warping $F_i^\top$ from $S^T I_i^L$ to $S^T I_0^L$
3: Compute $Y_i = F_i^T S^T I_i^L$
4: **End**
5: Compute $Q = \sum_{i=-T}^{T} F_i^T S^T S F_i$
6: Compute SR draft $Z = \sum_{i=-T}^{T} Q^{-1} Y_i$.

3.1. SR Draft Generation

In the problem of VideoSR, given an input sequence of LR frames $\{I_{-T}^L, \cdots, I_0^L, \cdots, I_T^L\}$, our goal is to recover the HR image I_0 corresponding to the LR reference frame I_0^L. Here T is the radius of the temporal neighborhood. The image formation model shows that each observed LR frame I_i^L is generated as

$$I_i^L = SKF_i I_0 + V_i, \tag{1}$$

where S, K and F_i are decimation, PSF kernel and warping operators for the i-th frame respectively. V_i is the additive noise. It is a complicated formation process since these variables and our target I_0 are all unknown in prior. This makes many previous methods take heavy computation to alternately estimate them in iterations, especially for the frame-wise motion operator F_i.

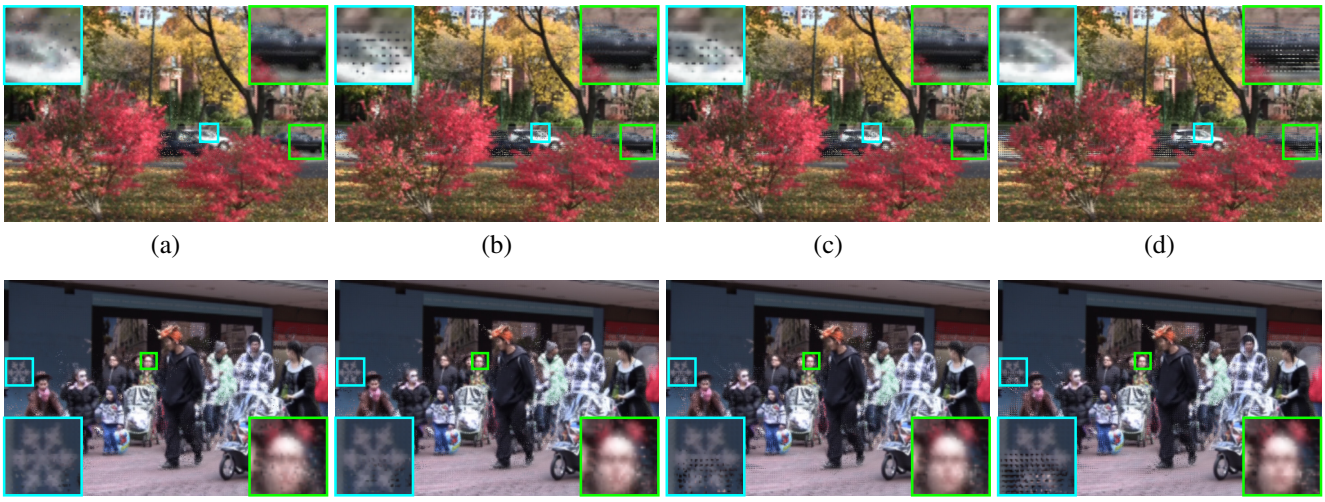
Our first component instead is to use simple methods to directly estimate SR drafts $Z = KI_0$. Because F_i involved in this process could be complex even locally, instead of producing one Z for each local window, we produce a set of them by varying the motion estimation strategy.

For fast processing, we modify the feedforward method [5] to generate a series of SR drafts for each window. As this is not our main contribution, we simply outline the process for each draft in Alg. 1. Here Y_i is the zero-upsampled and forward warped i-th frame and Q is diagonal. Thus inversion is equivalent to pixel-wise division. As Q is mostly singular in practice, we use bilinear interpolation in case of division by zero. Z is the reconstructed SR draft. Note that we omit the deconvolution step, which was included in the original method [5].

3.2. SR Draft-Ensemble and Its Visualization

The feedforward method in Alg. 1 runs quickly and has been evaluated in prior work. It was found that this process may not produce similarly good results as other methods proposed later due to its heavy reliance on motion estimation. It has no extra steps to update pixel correspondence when local displacement information is wrong.

Instead of relying on feedforward, we enhance this method by inputting more motion information. This pro-



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1. The 1st row shows the reconstructed blurred HR images by TV-ℓ_1 flow, where from (a) to (d), α is set to $0.005, 0.02, 0.025, 0.05$ respectively. The 2nd row shows the results of MDP flow, where λ is set to$5, 8, 14, 20$ respectively from (e) to (h).

cess makes it produce several SR Drafts D based on a set of motion estimates. In our method, we experiment with two robust optical flow algorithms. One is TV-ℓ_1 flow that is total variation regularized with the L1-norm data term [2, 11]. It is a common choice now for robust optical flow to reject outliers. The other is the motion detail preserving (MDP) flow [20] that has incorporated invariant feature correspondences and thus produces results different from TV-ℓ_1.

These two algorithms can be adjusted respectively on the weight of TV term α for TV-ℓ_1 and weight of smoothness term λ for MDP to produce motion estimates that are differently regularized. They yield a series of SR drafts Z after feedforward reconstruction in Alg. 1.

A few results are shown in Fig. 1. The two optical flow algorithms produce different results. For example, in the first row for the results of TV-ℓ_1, the zoom-in view of the white car-window is the best with $\alpha = 0.05$ while the black window is visually compelling with $\alpha = 0.02$ or 0.025. Similarly in the results of MDP in the second row, the snowflake information is the best with $\lambda = 5$ and the human face is good with $\lambda = 20$.

These results imply that regularization strength affects motion estimates. It is inherently necessary to have different motion estimation methods along with their various parameters to find the locally best estimates. By updating these factors in the feedforward reconstruction, we overcame the limitation that pixel correspondence cannot be updated in the original framework and obtain a family of SR drafts – each can be treated as an expert model for accurately estimating some kind of motion. The set of SR drafts is thus named SR draft-ensemble.

3.3. The Statistics of SR Draft Ensemble

To understand the statistical properties of our SR draft-ensemble, we conduct the following experiments. We collect 100 HR video sequences of resolution 800×1200 and generate the corresponding LR sequences following the general image formation procedure. Specifically, we apply a low-pass Gaussian filter with standard deviation $\sigma = 1.5$ to HR images and downsample them with a factor of 4. For each sequence, we treat the middle frame as the reference and compute all forward warping matrices based on TV-ℓ_1 optical flow under 20 different values of α (specified in our supplementary files), i.e., $\mathcal{A} = \{\alpha_i | i = 1, ..., 20\}$. Then we compute 20 corresponding SR drafts $\mathcal{Z} = \{Z_i | i = 1, ..., 20\}$ according to Alg. 1.

We randomly sample 1000 locations from every Gaussian filtered HR reference frame. The resultant ground-truth blurred HR reference frame is denoted as P^*. For each location, we collect patches with size 100×100 from the above SR drafts, denoted as $\mathcal{P} = \{P_i | i = 1, \ldots, 20\}$. Then we calculate the sum of squared difference (SSD) between P^* and P_i and find the minimal SSD value. Suppose P_m yields the minimal SSD. We say the corresponding α_m helps produce the best match for patch P^*.

We conduct two experiments to demonstrate important findings. The first experiment is to calculate the number of best-matches with respect to each α and plot the mean distribution in Fig. 2(e). This histogram shows all α are possibly the best in motion estimation for different regions in natural image sequences. Their distribution is not concentrated on only a few values, but instead rather uniform.

The second experiment is to validate that our set of α is enough in general for constructing high-quality blurred HR

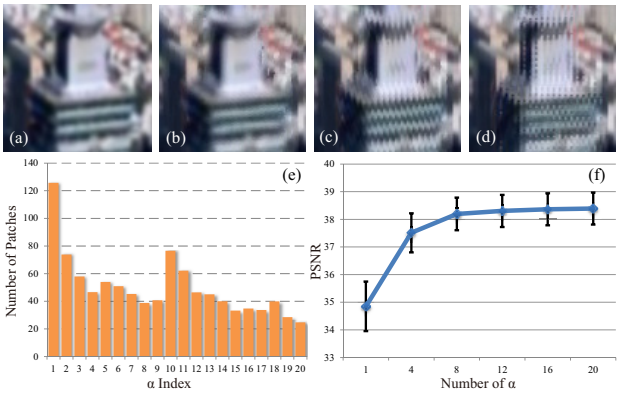


Figure 2. Statistics by varying motion estimation methods. (a) Ground truth blurred HR patch P^*. (b) $\sim$ (d) Corresponding SR drafts with $\alpha = 0.012, 0.08, 0.15$ and $SSD = 31.5, 55.9, 109.8$. (e) Histogram of the numbers of best-matches over 20 different α values. (f) Mean and standard deviation of PSNRs w.r.t. size k of the subset of $\mathcal{A}$.

images. In this experiment, we construct a few subsets of $\mathcal{A}$ containing the first k values of α in $\mathcal{A}$. For each sequence, with the given subset of $\mathcal{A}$, we reconstruct the best SR draft by picking up the best-match that yields the minimal SSD with the ground-truth blurred HR patch P^* at each position. Then we calculate the peak signal to noise ratio (PSNR) between the best SR draft and corresponding P^*. Fig. 2(f) plots the mean and standard deviation of PSNRs w.r.t. the size k of each subset.

This experiment manifests that results by applying these 20 values already contain SR drafts very close to the ground truth data under high PSNRs. If the best SR draft can be found for each location, the following reconstruction process can be achieved nicely. More α candidates only marginally increase the performance.

Now, the problem is how to find the optimal or near-optimal SR drafts for final image reconstruction. We employ a discriminative learning algorithm to find suitable candidates and accomplish the final reconstruction simultaneously.

4. Ensemble Learning via CNN

Our second main component in the system is SR draft ensemble learning to infer suitable drafts in a data-driven manner and apply the final deconvolution to reconstruct the HR images. This component is accomplished suitably through a deep convolutional neural network framework.

4.1. Motivation

With the multiple SR drafts – each can be regarded as one channel for the corresponding image – our input to this module is a multi-channel image. The goal to reduce this multi-channel image back to one-channel by inferring suitable SR drafts is however very challenging.

One possible solution is to treat this process as labeling where each patch is selected among a few using methods of discrete optimization under Markov random fields (MRFs). But this scheme needs to define appropriate potential functions. Hand-crafted costs may not work well for general natural videos because the process to compute the final SR draft from many candidates may not be simple selection. Instead highly nonlinear local operations could be involved.

Also, the solution through defining potential functions and optimizing them does not guarantee that their HR results can be successfully deconvolved for final visually-compelling reconstruction. An image that does not satisfy the convolution model easily generates visual artifacts, such as ringing, in the reconstructed HR image.

With these concerns, we resort to a CNN solution, which is found surprisingly capable to deal with these challenges. The advantages are threefold. First, the three-dimensional filter of CNN plays a role of continuous weight combination of multiple local spatial regions, which is beneficial for artifact removal. Second, our CNN framework is novel on concatenating two modules for SR draft construction and final reconstruction. The unified structure makes output SR drafts optimal w.r.t. the final clear image. Finally, the share-weight nature of CNN makes it effective in terms of representativeness than many classical models, such as pairwise MRF and run quickly during testing as the computation is only on a few convolution operations.

4.2. Network Architecture

As aforementioned, the input to our network is the c-channel image where each of the first $c-1$ channels corresponds to one image produced with one optical flow method. The last channel image is the bicubic-interpolated LR reference frame to lower bound computation in cases all employed optical flow algorithms fail due to extremely large motion.

The architecture of our CNN is shown in Fig. 3 where the output of the l-th layer is expressed recursively as

$$\begin{aligned} F^l(X) &= X && l = 0 \\ F^l(X) &= tanh(W^l * F^{l-1}(X) + B^l) && 1 \le l \le L-1 \\ F^l(X) &= W^l * F^{l-1}(X) + B^l && l = L \end{aligned} \quad (2)$$

Here the input sequence X is of size $h \times w \times c$, where h and w are the height and width and c denotes the channel number. W^l is the concatenation of n_l convolutional filters in the l-th layer and is of size $f_l \times f_l \times n_{l-1} \times n_l$. Here f_l is the spatial size of the filter, n_{l-1} is the number of filters in the last layer and B^l is the vector of bias with length n_l. The size of the output of the l-th layer is $h \times w \times n_l$. We use a *tanh* function as our nonlinear unit. In our network, the number of hidden layers is $L = 4$. The filter sizes are set as

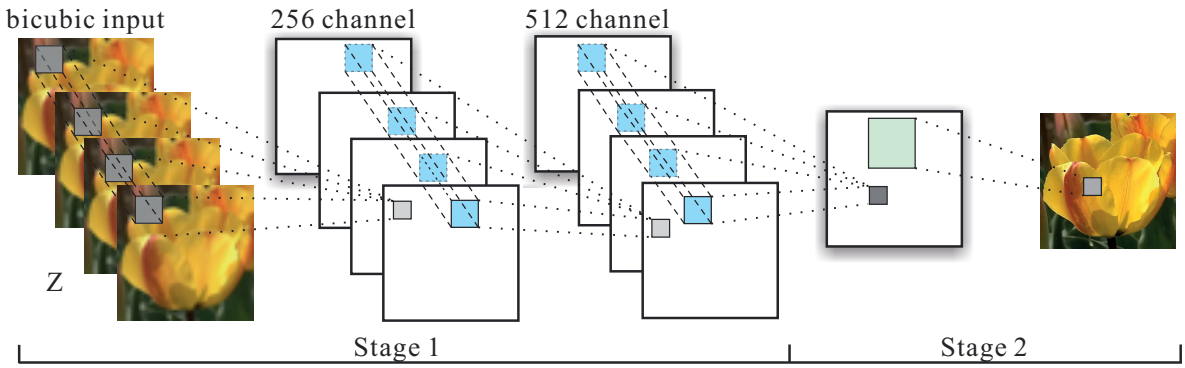


Figure 3. The two-stage architecture of our CNN framework. The input is a multi-channel image containing the set of reconstructed blurred HR images and the bicubic interpolated reference frame. The first and second stages aim at merging SR drafts and final deconvolution respectively.

$f_1 = 11$, $f_2 = 1$, $f_3 = 3$ and $f_4 = 25$. And the numbers of filters are $n_1 = 256$, $n_2 = 512$, $n_3 = 1$ and $n_4 = 1$.

To understand the design of our framework, we split the network into two parts. The first part consists of three convolutional layers and is expected to merge HR details from SR drafts in the region level. The architecture of this stage is similar to that of [4]. After processing the input sequence in the first a few layers, the output single-channel image is fed to the next part to perform deconvolution [10] and remove visual artifacts. Instead of adopting a large network as that in [10], we only use a 25×25 kernel, which is initialized by weights of an inverse kernel. We elaborate on parameter setting in Sec. 5.3.

4.3. CNN Training

For our CNN training, rather than adopting the ℓ_2 loss function as [4, 3], we exploit the ℓ_1 loss with total variation (TV) regularization, inspired by recent reconstruction methods [6, 12] to reduce visual artifacts. Moreover, the TV regularizer, which is imposed on the output of network, can be nicely incorporated into the back-propagation scheme.

Denoting the function represented by the network as $\mathcal{F}$, we minimize

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\|\mathcal{F}(X_i) - I_i\|_1 + \lambda\|\nabla\mathcal{F}(X_i)\|_1\right). \qquad (3)$$

Here the total number of training sequences is N. I_i is the i-th ground truth HR reference frame. $\|\nabla\mathcal{F}(X_i)\|_1$ is the aforementioned total variation term and λ is the corresponding regularization weight. To deal with the ℓ_1 norm, we use the Charbonnier penalty function $\Phi(x) = (x^2 + \varepsilon^2)^{1/2}$ for approximation. Here we empirically set $\lambda = 0.01$ and $\varepsilon = 0.001$. Learning the network is achieved via stochastic gradient descent with back-propagation.

5. Experiments

We build a VideoSR dataset[1] by collecting 160 video sequences from 26 high-quality 1080p HD video clips, which cover a wide variety of scenes and objects. These sequences are with complex non-rigid motion and present occlusion in different levels. For fair comparison with the method of [12], each sequence is trimmed to contain 31 consecutive frames where 15 forward and 15 backward neighboring frames are used to update the central one.

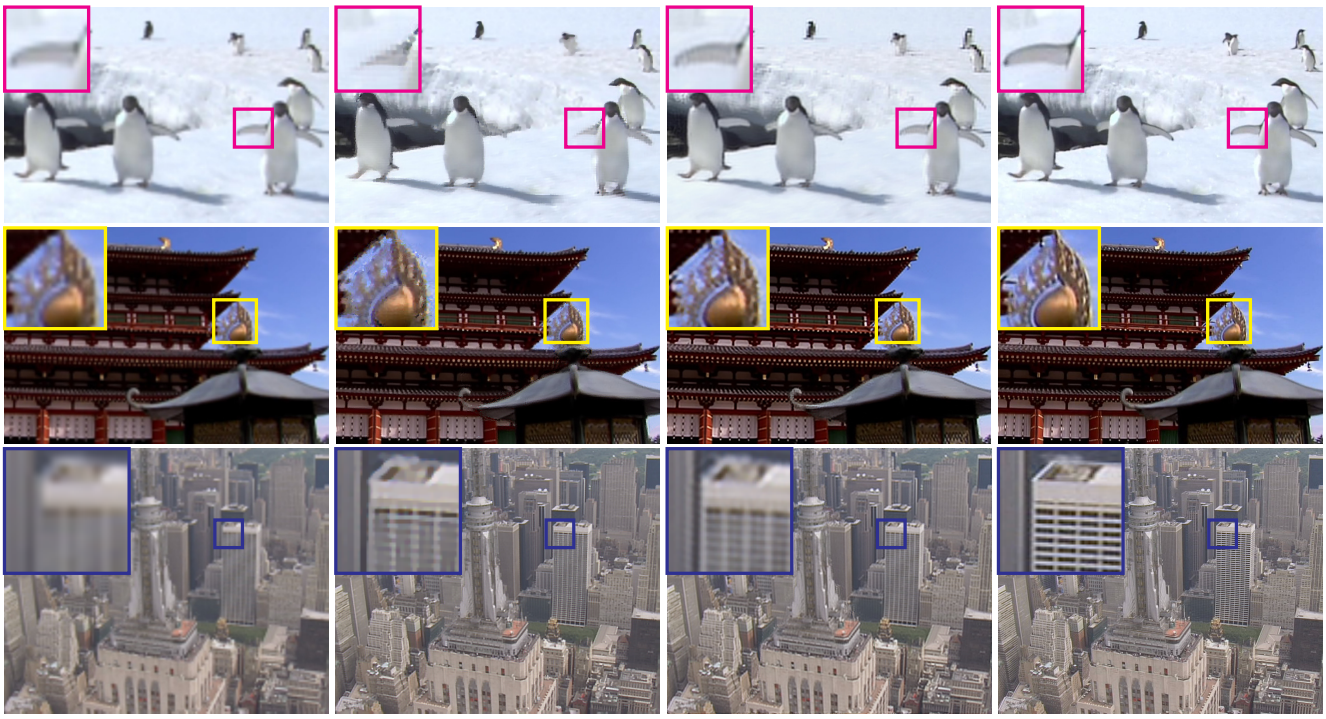
We generate the LR frames by first applying the low-pass Gaussian filter K to the HR frames for anti-aliasing and then downsampling them with a factor of 4. The setting of K is elaborated on in Sec. 5.1. In our dataset, we select 112 sequences for training and the remaining for testing. Due to limited memory, one thousand 100×100 patches are randomly sampled from SR drafts per sequence, thus resulting in a total of 112,000 training inputs. Moreover, to evaluate the generalization ability of our CNN model, we collect 40 real-world natural video sequences captured by cell phones and digital cameras with varying quality and containing a set of objects of flower, text, barcode, *etc*.

5.1. Implementation

We use a PC with an Intel Core i5 CPU and a NVIDIA K40 GPU. We implement our CNN based on the Caffe platform [8]. We use the publicly available implementation of TV-ℓ_1 [11] and MDP [20] optical flow algorithms. The convolutional filter in the second stage of our network is initialized by an inverse kernel, which has 25×25 spatial support, smaller than the one adopted in [21, 10].

To perform data augmentation on PSF K, we enlarge the training set by using multiple PSF kernels K to perform convolution. As suggested in [12], a PSF kernel for upscaling factor of 4 can be approximated by a Gaussian

[1]For dataset, code and more details, please visit our website (link in the front page).



(a) (b) (c) (d)

Figure 4. Comparison of synthetic sequences at a magnification factor of 4. (a) Bicubic interpolation of the reference frame. (b) Results of Bayesian Video SR [12]. (c) Our results. (d) Ground truth.

with standard deviation from 1.2 to 2.4. We thus adopt $K = 1.5, 1.8, 2.1$ in our experiments. The learning rates are 10^{-6} for the first convolutional layer and 10^{-7} for other layers. Smaller learning rates for the last layers are important for our CNN to converge. No weight decay is employed during our experiments. We train our model for approximately 3×10^7 back-propagations. For RGB sequences, we treat color channels separately during training and testing.

5.2. Validation of Our Network

We first validate our method on both generated and real-world data. For generated LR data, we compare results with state-of-the-art Bayesian adaptive video SR (BayesSR) [12] in Fig. 4. More are contained in the project website. For the example *penguin* shown in the first row of Fig. 4, the wing part undergoes extremely large motion. While for the cases of *temple* and *city* in bottom two rows, similar difficulties occur in the adornment and facade respectively, easily leading to visual artifacts or erroneous reconstruction near the boundary of moving objects.

Our results are more natural due to sufficient SR drafts that are used and the nonlinear learning procedure by CNN to further reduce visual artifacts. We calculate the PSNR and SSIM [19] values and list them in Table 1. Our results are reasonable under these quantitative measures. We note BayesSR incorporates advanced optimization and sparse models, and thus requires much more computation.

Table 1. Comparison based on PSNR & SSIM.

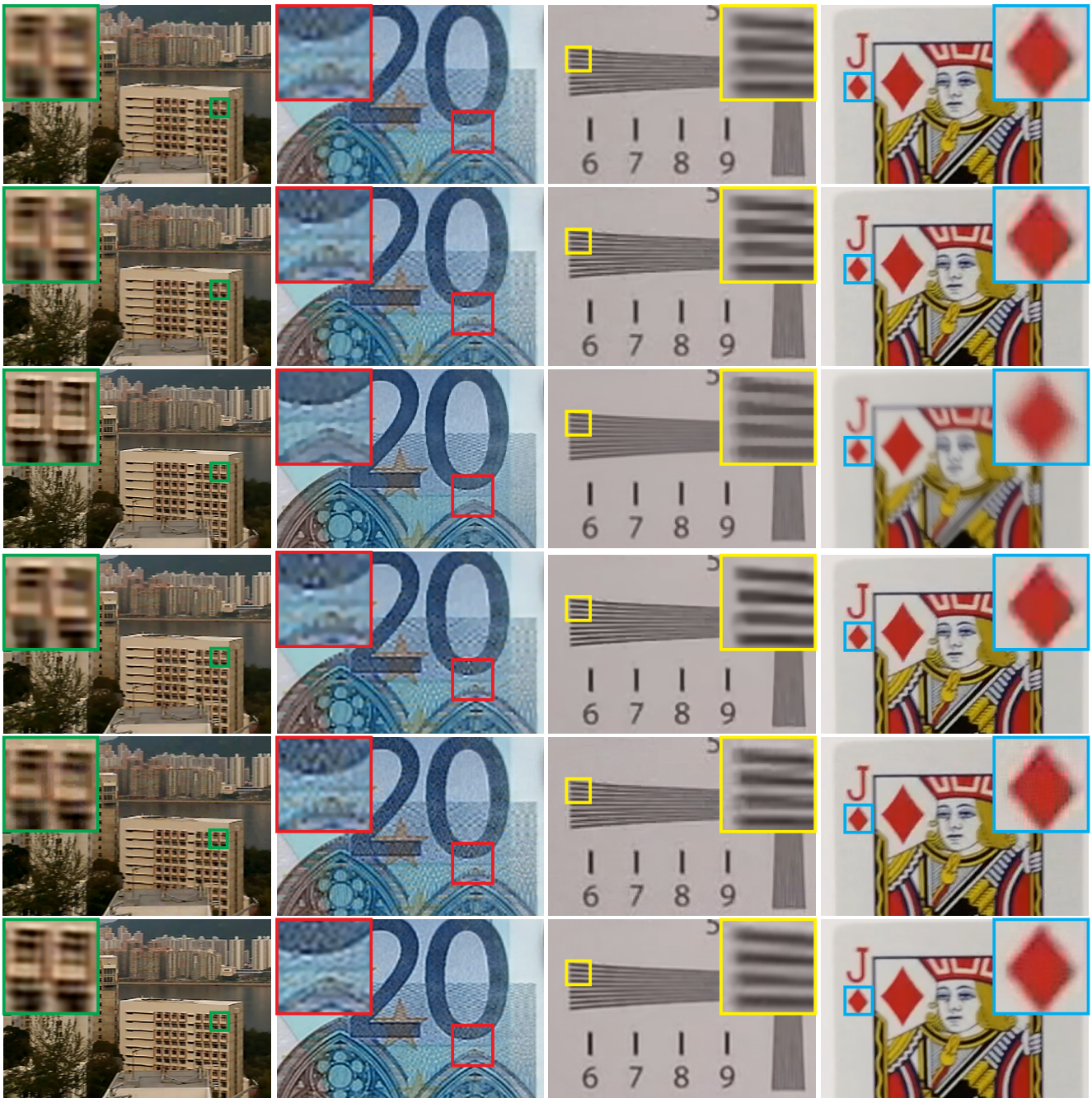
PSNR	BayesSR	Ours	SSIM	BayesSR	Ours
penguin	29.53	31.87	-	0.9451	0.9483
temple	29.01	30.23	-	0.9375	0.9504
city	25.49	24.89	-	0.7979	0.7610

For the challenging real sequences, we compare our method with commercial software Video Enhancer (V1.9.10) [1], fast video upsampling [15], 3DKR [16] and BayesSR. We show results of 4 sequences captured by us with different image qualities in Fig. 5.

For example, for the *building-window* example in column (a), Our result contains many details. For the examples of *euro* in column (b), our result not only generates sharp edges but also reconstructs a level of texture. In column (c), our result is with reduced artifacts. The last input low-resolution *poker* image is a bit motion blurred. Our method even deals with this problem and largely suppresses the JPEG artifacts contained in the input videos.

We also compare our method with the single-image one [3] and sharpened bicubic upsampling in Fig. 6. Our method produces more details, manifesting that useful information indeed exists in multiple frames. The learning-based method [3] could generate less structures than ours.

We apply our system to a low-quality surveillance video



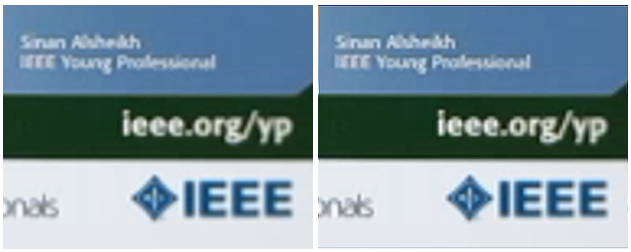
(a) (b) (c) (d)

Figure 5. Comparison on real natural sequences at a magnification factor of 4. The results from the first to the last rows are from bicubic interpolation, 3DKR [16], BayseSR [12], Fast Video Upsampling [15], VideoEnhancer [1] and ours.

sequence, which is trimmed to 31 LR frames of resolution 288×352. The result with upscaling factor 4 is shown in Fig. 7. Note that this video contains very strong JPEG artifacts, which easily fail existing algorithms. Contrarily, our system is free of tuning parameters in the testing phase. Our result in (d) is with genuine higher resolution than input frames from (a)-(c) and contains less artifacts. More results with close-ups are contained in the project website.

5.3. More Analysis

We now evaluate importance of parameter setting and initialization. We denote our method as "std" with the in-



(a) Input (b) Sharpened Bicubic

(c) Results of [3] (d) Ours

Figure 6. Comparisons with single-image methods at a magnification factor of 4.

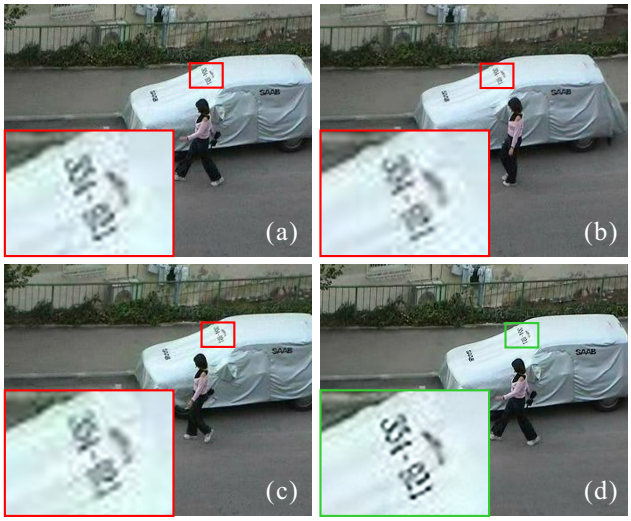


Figure 7. Results of a surveillance video. (a)-(c) are the bicubic upsampling results of frames 16, 20, and 30. (d) is our HR result of the 16-th frame with upscaling factor 4.

Table 2. Comparison of different parameter setting.

penguin	std.	w/o inv.	5×5	21×21	ℓ_2
PSNR	31.87	28.81	31.41	31.29	30.19
SSIM	0.9483	0.9428	0.9453	0.9521	0.9460
temple	std.	w/o inv.	5×5	21×21	ℓ_2
PSNR	30.23	29.17	28.86	27.76	27.28
SSIM	0.9504	0.8860	0.9292	0.8934	0.8993

verse kernel initialization in the last layer, 11×11 kernel size in the first layer and with the TV-ℓ_1 loss function. First, we show the effect of weight initialization in the last layer by the inverse kernel. To rule out the effect of other factors, we fix other-layer initialization and parameter values. The results are listed in Table 2 (the 1st & 2nd columns), showing that inverse kernel initialization generally improves results compared to not using such a scheme, denoted as "w/o inv.", in terms of PSNR and SSIM.

Then we evaluate different sizes of the convolutional kernel in the first layer. Again, we fix other parameters and only vary the kernel size. The results are reported in Table 2 (the 3rd & 4th columns for kernel sizes of 5×5 & 21×21). Our "std." system configuration is with the 11×11 kernel size. It produces results comparable to using other settings, showing that our system is not that sensitive to this parameter. We note the computational cost of using 11×11 kernels is much less than the choice of 21×21.

Finally, we evaluate the loss function and compare the TV-ℓ_1 form included in our "std." configuration with traditional ℓ_2 loss under the same name in Table 2. TV-ℓ_1 is consistently better than the ℓ_2 loss in terms of PSNR and SSIM. This may be partly due to higher robustness of TV-ℓ_1 to outliers.

5.4. Running Time

Once our system is trained, it is very efficient to execute in testing. We record the running time of our method on an input LR sequence of 31 frames with size 120×180 under a magnification factor of 4. The overall time cost of our method splits into two main parts. The first is for computing the forward warping matrix by the TV-ℓ_1 flow – it is about $58s$ for one image. The other part is for the test of our CNN, which is about $0.2s$ for each color channel. If we sequentially compute multiple SR drafts, total time used is $500s \approx 8min$, which is less than the 2-hour reported in [12]. We further accelerate it to less than 60 seconds using parallelization.

6. Conclusion

In this paper, we have proposed a SR draft-ensemble framework, which exploits CNN to solve the VideoSR problem. We observe that SR drafts obtained through simple feedforward reconstruction procedures by varying motion estimation setting, contain generally sufficient information for estimating the final HR image. Based on this finding, we resort to CNN to integrate the reconstruction and deconvolution steps. Our framework produces decent results on many sequences. Future work will be to further enhance the ability of this framework to handle even higher super-resolution ratios and incorporate SR draft generation into one unified network.

Acknowledgements

This research is supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant number 412911.

References

[1] Video Enhancer. http://www.infognition.com/videoenhancer/, 2010.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*. 2004.

[3] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014.

[4] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, 2013.

[5] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. *IEEE TIP*, 10(8):1187–1193, 2001.

[6] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE TIP*, 13(10):1327–1344, 2004.

[7] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP*, 53(3):231–239, 1991.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[9] S. Kim, N. K. Bose, and H. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(6):1013–1027, 1990.

[10] X. Li, R. Jimmy, L. Ce, and J. Jiaya. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014.

[11] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[12] C. Liu and D. Sun. On bayesian adaptive video super resolution. *IEEE TPAMI*, 2013.

[13] Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, and E. Wu. Handling motion blur in multi-frame super-resolution. In *CVPR*, 2015.

[14] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE TIP*, 18(1):36–51, 2009.

[15] Q. Shan, Z. Li, J. Jia, and C.-K. Tang. Fast image/video upsampling. *ACM TOG*, 27(5):153, 2008.

[16] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super-resolution without explicit subpixel motion estimation. *IEEE TIP*, 18(9):1958–1975, 2009.

[17] R. Tsai and T. S. Huang. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing*, 1(2):317–339, 1984.

[18] H. Ur and D. Gross. Improved resolution from subpixel shifted pictures. *CVGIP*, 54(2):181–186, 1992.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[20] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE TPAMI*, 34(9):1744–1757, 2012.

[21] L. Xu, X. Tao, and J. Jia. Inverse kernels for fast spatial deconvolution. In *ECCV*, 2014.