# VARIATION LEARNING GUIDED CONVOLUTIONAL NETWORK FOR IMAGE INTERPOLATION

*Wenhan Yang*[1]*, Jiaying Liu*[1*]*, Sifeng Xia*[1]*, Zongming Guo*[1,2]

[1]Institute of Computer Science and Technology, Peking University, Beijing, P.R. China
[2]Cooperative Medianet Innovation Center, Shanghai, P.R. China

## ABSTRACT

In this paper, we propose a variational learning model that effectively exploits the structural similarities for image representation, and construct a deep network based on this model for image interpolation. Based on the local dependency, our learning model represents an image as the three-dimensional features. Besides two coordinate dimensions, an additional *neighboring variation* dimension is added to encode every pixel as the variation to its nearest low-resolution pixel by the local similarity. This added dimension lowers the risk of over-fitting for learning approaches and constructs abundant structural correspondences for inferring the missing information lost in image degradation. Then, this three-dimensional features are naturally modeled, extracted and refined by an end-to-end trainable recurrent convolutional network for image interpolation. Comprehensive experiments demonstrate that our method leads to a surprisingly superior performance and offers new state-of-the-art benchmark.

***Index Terms***— Variation Learning, deep learning, image interpolation

## 1. INTRODUCTION

Image interpolation is a fundamental research topic that reconstructs a high-resolution (HR) image from one of its down-sampled low-resolution (LR) versions by estimating all missing pixels during the down-sampling process. Util now, various interpolation methods could be categorized into three classes: *polynomial-based methods*, *geometry-guided methods* and *learning-based methods*.

*Polynomial-based methods*, such as Bilinear and Bicubic methods [1], interpolate images by convolving neighboring pixels with the fixed kernels. They have a relatively low computational complexity but their results contain noticeable artifacts (*e.g.* blurring, ringing, jaggies and zippering) and unnatural representations of edges.

To utilize local structural information and achieve visually pleasing results, *geometry guided methods* are proposed, including explicit geometry and implicit methods. Explicit geometry guided methods detect the geometric features, *e.g.* edges and local covariances, and adjust the interpolation lattice or directions dynamically [2, 3]. To reduce the risk of inaccurate edge detection, several methods [4, 5] are proposed to model edges with soft models to fuse the information from multiple edge directions.

The other kind of geometry guided methods – implicit geometry guided methods – embed statistical geometric information into an optimization function and obtain an adaptive filter that maximizes the function [6–10]. This joint modeling for LR and HR pixels describes their correlations more intrinsically and makes the implicit geometry guided methods achieve superior performance.

Polynomial-based and implicit geometry guided methods predict HR pixels via a hand-crafted pattern or an adaptive filter estimated based on local image structures. In these methods, the effective knowledge from external natural images have not been utilized for image modelling, which leaves space for further improvement of image interpolation.

Later on, *learning-based interpolation approaches* have been attracting much attention. Learning the mapping from LR pixels to missing HR pixels from a large collected data set, these methods achieve very promising results with rather high computational efficiency. In [11, 12], a joint model with sparse dictionary learning, nonlocal patch prior and autoregressive model is constructed to effectively capture and further make full use of the nonlocal similarity to facilitate the HR pixel estimation. However, only utilizing the internal prior within the LR image leads to performing poorly in the non-repetitive regions. In [13], random forests are built to partition the natural image patch space into numerous subspaces. In each subspace, a linear regression model is exploited to transform the LR image patch into an HR one. However, the subspace partition and local regression are optimized separately and the local regression is based on a linear model. These two factors limit its modeling capacity to model the complex mapping relationship.

Recently, a series of successful deep learning methods have emerged, *e.g.* image denoising [14, 15], completion [16], super-resolution (SR) [17, 18]. Due to the different configurations of image SR and interpolation, existing image SR works cannot be directly applied for image interpolation, where the sampling matrix or observation matrix is usually canonical matrix and leads to discontinuous degraded signals. Both these huge successes and the dilemma for image interpolation guide us to explore constructing a deep network based on our proposed variation image representation for image interpolation.

In this paper, we propose a three-dimensional variation image representation that well fits for solving the image interpolation problem. The representation encodes a pixel as its variation value to the corresponding top-left neighboring LR pixel and extends to a three-dimensional representation. This effective modeling for the local dependency decreases the correlation in the estimated mapping from LR to HR images, and makes the learned priors more effective, as well as increases structural correspondences. Based on this model, we propose a variation learning network (VLN) to estimate the variation value between the HR pixel and its nearest LR pixel. Extensive experimental results demonstrate that, our method leads to a surprisingly superior performance than other methods and offers new state-of-the-art benchmark.

The rest of this paper is organized as follows. Section 2 in-

*Corresponding author. This work was supported by National Natural Science Foundation of China under contract No. U1636206 and Culture Development Funding under Grant No.2016-288.

troduces our three-dimensional variation image model and briefly compares it with the popular residue learning model. In Section 3, we attempt to design a deep network for image interpolation. Experimental results are presented in Section 4. Finally, concluding remarks are given in Section 5.

## 2. THE PROPOSED THREE DIMENSIONAL VARIATION IMAGE MODEL

Learning-based approaches [11–13] aim to solve the image interpolation by estimating the inverse mapping

$$\mathbf{x} = f_a(\mathbf{y}), \tag{1}$$

for image degradation $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{v}$, where $\mathbf{D}$ is a decimal matrix and $\mathbf{v}$ is the random noise term. These approaches that directly model the mapping between $\mathbf{y}$ and $\mathbf{x}$ usually suffer from several drawbacks: 1) the learned model over-fits to the regularity between low-frequency parts of image signals and high-frequency details are neglected; 2) the priors are imposed on the whole $\mathbf{x}$, thus the method is hard to learn useful guidances for recovering the high frequency image signal; 3) useful structural correspondences in the high frequency domain may be neglected when directly modeling $\mathbf{x}$.

To overcome the aforementioned drawbacks, we propose a novel image model – three-dimensional variation image representation. Through embedding the local redundancy, we can remove the auto-correlation of $\mathbf{x}$ and $\mathbf{y}$ as well their correlation, and to construct more useful structural correspondences within an image.

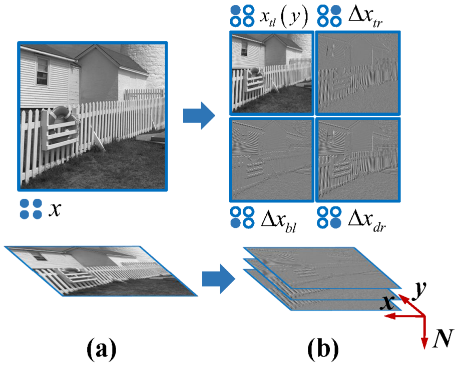


**Fig. 1**. The three-dimensional variation representation based on local redundancy.

The aforementioned image representation is intuitively shown in Fig. 1. Based on the local redundancy, a local pixel could be represented by the summation of one of its nearest neighbors and a very small difference value, called the variation. Correspondingly, an HR pixel could be represented by the summation of the top-left LR pixel in the corresponding $2 \times 2$ non-overlapping patch (for convenience, we take $2\times$ enlargement as example all through this paper but note that, our method is general to apply for other times enlargement and our method is also evaluated in $3\times$ enlargement in our experiment) and a difference value between the LR and HR pixels. This split operation removes much of auto-correlation within the image. Equally, an HR image $\mathbf{x}$ is split into four parts $\mathbf{x}_{tl}$, $\Delta\mathbf{x}_{tr}$, $\Delta\mathbf{x}_{bl}$ and $\Delta\mathbf{x}_{br}$.

$$\begin{aligned} \mathbf{x}_{tl} &= y, \\ \Delta\mathbf{x}_{tr} &= \mathbf{x}_{tr} - \mathbf{x}_{tl}, \\ \Delta\mathbf{x}_{bl} &= \mathbf{x}_{bl} - \mathbf{x}_{tl}, \\ \Delta\mathbf{x}_{br} &= \mathbf{x}_{br} - \mathbf{x}_{tl}, \end{aligned} \tag{2}$$

where $\mathbf{x}_{tl}$, $\mathbf{x}_{tr}$, $\mathbf{x}_{bl}$ and $\mathbf{x}_{br}$ are the sub-images consisting of the top-left, top-right, bottom-left and bottom-right pixels extracted from every $2 \times 2$ non-overlapping patches. The last three terms stack as a tensor as shown in Fig. 1 where two axises signify the locations, and another axis signifies the neighbors domain.

Thus, $\mathbf{x}$ is reformulated as

$$\mathbf{x} = \mathbf{x}_\epsilon + \Delta\mathbf{x}, \tag{3}$$

where $\mathbf{x}_\epsilon$ denotes the top-left pixel (in fact, a nearest LR pixel) in every $2 \times 2$ non-overlapped patch. For convenience, $\Delta\mathbf{x}$ is defined as (2) with $\Delta\mathbf{x}_{tl} = 0$. In image interpolation, $\mathbf{x}_\epsilon$ is given (equivalent to $\mathbf{y}$), thus we can focus on estimating $\Delta\mathbf{x}$ which naturally leads to a new learning paradigm

$$\mathbf{x} = f_r(\mathbf{y}) + \mathbf{x}_\epsilon. \tag{4}$$

$f_r(\cdot)$ is the learned inverse recovery process to estimate $\mathbf{x} - \mathbf{x}_\epsilon$ from $\mathbf{y}$.

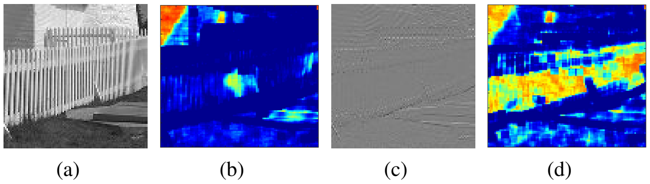

(a) (b) (c) (d)

**Fig. 2**. Patch repetitiveness for several image models.

Our novel proposed image representation (4) has three major advantages compared with (1):

- *Fitting to high frequency image signals*. Based on local dependency, low frequency part is removed, and the regularity between the high frequency is focuses on.
- *Direct priors*. The priors are imposed on the missing part of $\mathbf{x}$ now, which benefits learning useful guidance for recovering the high frequency part of the image signal.
- *Plenty of structural correspondences*. We also compare (4) and (1) from the perspective of structural correspondences on their patch repetitiveness that measures the potential redundancy within an image. We calculate it via mean squared error (MSE) for the most similar patches of each $5\times5$ patch. We first search the top-10 similar patches based on MSEs across the whole image for each patch. Then, the average MSE is converted into a probability based on Gaussian function. As shown in Fig. 2, the subfigures (b) and (d) are the heat maps for the patch repetitiveness of (a) – an normal 2D sub-image in (1), that for the patch repetitiveness of (c) – the difference image – in the variation space in (4) and that for the patch repetitiveness of (d) in the variation space in (4), respectively. In these heat maps, the colors from red to blue signify the decrease of patch repetitiveness values. Compared with (b), the extensions (d) significantly increase the patch repetitiveness.

We also simply compare the variational learning and the popular residue learning [19,20] for image upsampling. Although residue learning achieves rather impressive results in image SR, these well developed approaches cannot be directly applied for image interpolation. The similar conclusion has been mentioned in [11]. Compared the blur matrix in image SR, the identity matrix $\mathbf{H}$ in image interpolation does not cut-off the signal at a certain frequency band in the sampling. Natural images are usually not band-limited due

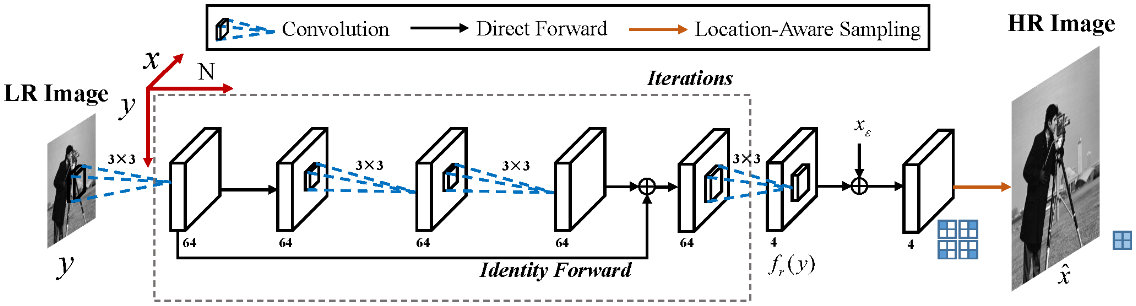
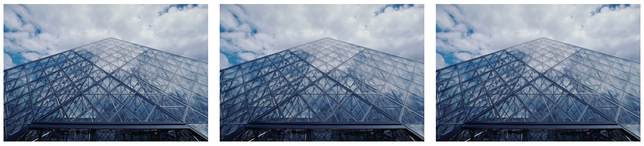


**Fig. 4**. The architecture of our proposed variation learning network (VLN) for image interpolation.



(a) Original (b) VLN, 28.05dB (c) VDSR, 24.13dB

**Fig. 3**. Deep learning-based image interpolation and their PSNR results.

to many sharp structures, thus the measured signals are usually spatially discontinuous compared with that from image SR degradation. This difference makes deep-learning based SR methods, which rely on the smoothness of image signals, fail to handle image interpolation. Fig. 3 shows an example, the above mentioned deep-learning image SR methods – VDSR (retrained with interpolation degradation) – results in blurring and artifacts of ringings and zippers.

## 3. VARIATION LEARNING NETWORK FOR IMAGE INTERPOLATION

We turn our variation image model (4) into an end-to-end trainable variation learning network (VLN) for image interpolation. In general, our network takes a recurrent convolutional structure that performs a progressive recovery route and models the first three dimensions naturally. Firstly, the network extracts LR features by the first convolution. Secondly, the features are transformed and enhanced from the LR space to HR space iteratively. Thirdly, the variation map is reconstructed by the last convolution on the enhanced features. Fourthly, the proposed network finally combines the pixel variation and the corresponding top-left pixel in each non-overlapped patch. Finally, the HR image is reconstructed by a location-aware up-sampling layer. The layer transforms the pixels of four maps (LR image and HR sub-images) into the HR image prediction.

Specifically, we illustrate each part in formulation:

1. *Feature extraction and reconstruction*. The front-end convolution extracts features $\mathbf{f}_{\text{in}}^1$ from the LR image, and the penultimate convolution layer reconstructs the HR difference maps from features $\mathbf{f}_{\text{out}}^K$. The relation between $\mathbf{f}_{\text{in}}^1$, $\mathbf{f}_{\text{out}}^K$ and the other part of the network is

$$\mathbf{f}_{\text{in}}^1 = \max(0, \mathbf{W}_{\text{extract}} * \mathbf{y} + \mathbf{b}_{\text{extract}}), \tag{5}$$

$$\Delta\mathbf{x} = [\Delta\mathbf{x}_{\text{tl}}, \Delta\mathbf{x}_{\text{tr}}, \Delta\mathbf{x}_{\text{dl}}, \Delta\mathbf{x}_{\text{dr}}] = \mathbf{W}_{\text{rect}} * \mathbf{f}_{\text{out}}^K + \mathbf{b}_{\text{rect}}, \tag{6}$$

where $\mathbf{y}$ is the input LR image, and $\mathbf{W}_{\text{extract}}$ and $\mathbf{b}_{\text{extract}}$ denote the filter parameter and basis of the first convolution layer – feature extraction layer, respectively. As shown in Fig. 4, $\Delta\mathbf{x}_{\text{tl}}$, $\Delta\mathbf{x}_{\text{tr}}$,$\Delta\mathbf{x}_{\text{dl}}$ and $\Delta\mathbf{x}_{\text{dr}}$ are the estimated top-left, top-right, down-left and down-right values in every $2 \times 2$ non-overlapping patch by the reconstruction layer. $\mathbf{W}_{\text{rect}}$ and $\mathbf{b}_{\text{rect}}$ denote the filter parameter and basis of the reconstruction layer. $\Delta\mathbf{x}_{\text{tr}}$,$\Delta\mathbf{x}_{\text{dl}}$ and $\Delta\mathbf{x}_{\text{dr}}$ are then added with the LR image respectively to generate the corresponding HR pixels $\hat{\mathbf{x}}$ (a four channel feature map) in these locations, and $\Delta\mathbf{x}_{\text{tl}}$ equals to a zero map due to the location correspondences in the degradation.

2. *Progressive feature enhancement*. Let $\mathbf{f}_{\text{in}}^k$ denote the input feature map for the recurrent sub-network at the $k$-th recurrence. The output feature map of the recurrent sub-network, $\mathbf{f}_{\text{out}}^k$, is progressively updated as follows:

$$\begin{aligned} \mathbf{f}_{\text{out}}^k &= \max\left(0, M^k + +\mathbf{f}_{\text{in}}^k\right), \\ M^k &= \left(\mathbf{W}_{\text{mid}}^k * \mathbf{f}_{\text{mid}}^k + \mathbf{b}_{\text{mid}}^k\right), \\ \mathbf{f}_{\text{mid}}^k &= \max\left(0, \mathbf{W}_{\text{in}}^k * \mathbf{f}_{\text{in}}^k + \mathbf{b}_{\text{in}}^k\right), \end{aligned} \tag{7}$$

where $\mathbf{f}_{\text{in}}^k = \mathbf{f}_{\text{out}}^{k-1}$ is the output features by the recurrent sub-network at $(k-1)$-th time step. $\mathbf{W}_{\text{in}}^k$ and $\mathbf{b}_{\text{in}}^k$ are the filter parameter and basis of the first convolution in the $k$-th iteration. $\mathbf{W}_{\text{mid}}^k$ and $\mathbf{b}_{\text{mid}}^k$ are the filter parameter and basis of the second convolution in the $k$-th iteration. The by-pass connection here is between $\mathbf{f}_{\text{in}}^k$ and $\mathbf{f}_{\text{out}}^k$. The feature map $\mathbf{f}_{\text{out}}^k$ can be viewed as the recovered $k$-th layer details of the feature maps.

3. *Location-aware up-sampling.* The location-aware up-sampling layer transforms four variation maps back to the HR lattice as

$$\hat{\mathbf{x}}_{\text{up}}(i, j, c) = \hat{\mathbf{x}}(\lfloor i \times s \rfloor, \lfloor j \times s \rfloor, c), \tag{8}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, $s$ signifies the scale of one convolution path, $p$ signifies the group of the output number. Here $s$ is set to 1/2 and 1/3 with the scaling factor 2 and 3 reflectivity. $i$ and $j$ denote the spatial location and $c$ denotes the channel number. ‘up’ denotes the output result is up-sampled to the HR lattice. $p$ is calculated as follows,

$$\begin{aligned} p =& (i - \lfloor i \times s \rfloor - 1) \times 1/s \\ &+ (j - \lfloor j \times s \rfloor) + 1. \end{aligned} \tag{9}$$

4. *Network training.* Let $\mathbf{F}(\cdot)$ represent the learned network to recover the HR image $\mathbf{x}$ based on the input LR image $\mathbf{y}$. We use $\mathbf{\Theta}$ to collectively denote all the parameters of the network,

$$\begin{aligned}\mathbf{\Theta} = \{&\mathbf{W}_{\text{extract}}, \mathbf{b}_{\text{extract}}, \mathbf{W}_{\text{in}}, \mathbf{b}_{\text{in}}, \\ &\mathbf{W}_{\text{mid}}, \mathbf{b}_{\text{mid}}, \mathbf{W}_{\text{rect}}, \mathbf{b}_{\text{rect}}\}\,.\end{aligned} \tag{10}$$

Given $n$ pairs of HR and LR images $\{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{n}$ for training, we adopt the following joint MSE to train the HR image estimation network parameterized by $\mathbf{\Theta}$:

$$L(\mathbf{\Theta}) = \frac{1}{n}\sum_{i=1}^{n}(||\mathbf{F}\left(\mathbf{Y}_i, \mathbf{X}_i; \mathbf{\Theta}\right) - \mathbf{X}_i||^2. \tag{11}$$

It is optimized by scholastic gradient descend (SGD).

## 4. EXPERIMENTAL RESULTS

The proposed algorithm is compared with conventional polynomial-based Bicubic interpolation method and six state-of-the-art interpolation algorithms, including soft autoregressive interpolation (SAI) [8], similarity modulated block estimation (SMBE) [21], consistent segment adaptive gradient angle interpolation (CSAGA) [22], adaptive general scale interpolation (AGSI) [23], nonlocal autoregressive modeling (NARM) [11] and fast interpolation via random forest (FIRF) [13]. We compare the proposed VLN with recent interpolation methods on three benchmark datasets: *Interp15*, *Interp18* and *Urban12*, with the scaling factor 2. The three datasets contain 15, 18 and 12 images respectively. Among them, the images in *Interp15* are from the Kodak and USC-SIPI image databases. *Interp18* is used for the evaluation in [13]. *Urban12* includes 12 urban landscapes images from *Urban* [24] dataset, that contains the images with many regular repetitive building patterns. Our dataset and all experimental results are online available[1]. Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity index (SSIM) [25] are used to evaluate the experimental results.

We trained our VLN with a training set containing 591 images, consisting of BSD500 [26] and 91 images in [27]. They were cropped into $40 \times 40$ input and $80 \times 80$ output patches. These images were decomposed into around 500,000 sub-images using a stride of 20 with the data augmentation of flipping and rotation. Our VLN was trained on Caffe platform [28] via stochastic gradient descent (SGD) with standard back-propagation. The momentum is set as 0.9, the initial learning rate as a fixed value 0.001 for front-end layers and 0.00001 for the penultimate layer (before the fixed location up-sampling layer) during the training. The learning rate is dropped when reaching 250,000 iterations by a factor of 10. The batch size is set as 64. We allowed at most 300,000 back-propagations, which spent about 7 hours on a single GPU – GTX 1080.

The objective evaluation results are shown in Tables 1 and 2. The results clearly demonstrated that our method consistently outperforms those well-established baselines with significant performance gains. For *Set15*, *Set18* and *Urban12*, our method – VLN (20L) – achieves better performance than FIRF with gains of 0.42, 0.52 and 1.15dB in PSNR, respectively. We also present some visual results in Figs. 5 to investigate all the methods intuitively. The results clearly show the significant superiority of our method to other baselines. From the figures, other compared methods generate the results with severe artifacts. The superiority obviously appears in the regions of the axises in *Bicycle*, the repetitive patterns of house wall in *Lighthouse* and the window boundaries in *080*. To further evaluate the generality of our method, we also compare more challenging 3× enlargement. The compared approaches include Bicubic and NARM. Other methods do not support this scaling factor. From Table 2, our method achieves significantly superiors performance than NARM, with gains more than 0.5dB in PSNR.

**Table 1**. The average PSNR (dB) Results on *Set15*, *Set18* and *Urban12* in 2× enlargement.

| Image | Bicubic | SAI | SMBE | CSAGA | AGSI |
|---|---|---|---|---|---|
| *Set15* | 28.81 | 29.19 | 29.32 | 29.20 | 29.06 |
| *Set18* | 28.82 | 29.44 | 29.47 | 29.29 | 29.33 |
| *Urban12* | 23.29 | 23.86 | 24.01 | 23.96 | 23.88 |
| Image | NARM | FIRF (1) | FIRF (5) | VLN (10L) | VLN (20L) |
| *Set15* | 29.47 | 29.58 | 29.81 | 30.19 | **30.23** |
| *Set18* | 29.75 | 29.98 | 30.11 | 30.55 | **30.63** |
| *Urban12* | 23.98 | 24.85 | 24.97 | 25.90 | **26.12** |

**Table 2**. The average PSNR (dB) Results on *Set15*, *Set18* and *Urban12* in 3× enlargement.

| Image | Bicubic | NARM | VLN (20L) |
|---|---|---|---|
| *Set15* | 25.68 | 26.60 | **27.12** |
| *Set18* | 25.24 | 25.31 | **26.72** |
| *Urban12* | 20.49 | 22.00 | **22.50** |

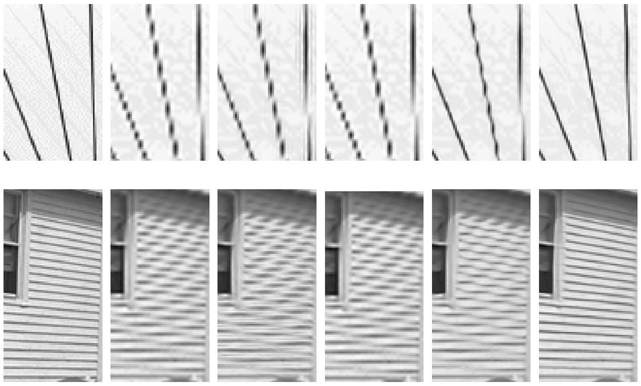


**Fig. 5**. Visual comparison between different algorithms in 2× enlargement. From top to bottom: *Bicycle* in *Set18*, *Lighthouse* in *Set15*, *080* in *Urban12*. From left to right: HR, Bicubic, SAI, NARM, FIRF, Proposed.

## 5. CONCLUSIONS AND DISCUSSIONS

In this paper, we propose a novel three-dimensional variation image representation and develop a variation learning network. The representation focuses on the correlation between high-frequency image signals, imposes priors directly on the variation between the LR and HR images and includes a number of structural correspondences. Owning to these benefits, a variation learning network is constructed for image interpolation. Embedding the local dependency, the network transforms and enhances image signals from LR space to HR space gradually, leading to a surprisingly superior performance than previous methods and offering the new state-of-the-art.

[1] http://www.icst.pku.edu.cn/struct/Projects/VLN.html

## 6. REFERENCES

[1] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, Dec 1981.

[2] Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 889–900, Apr 2007.

[3] C. M. Zwart and D. H. Frakes, "Segment adaptive gradient angle interpolation," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 2960–2969, Aug 2013.

[4] M. Li and T. Q. Nguyen, "Markov random field model-based edge-directed image interpolation," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1121–1128, Jul 2008.

[5] Lei Zhang and Xiaolin Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2226–2238, Aug 2006.

[6] Xin Li and M.T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, October 2001.

[7] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760–2768, Oct 2011.

[8] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 887–896, Jun 2008.

[9] W. Yang, J. Liu, M. Li, and Z. Guo, "Isophote-constrained autoregressive model with adaptive window extension for image interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2016.

[10] M. Li, J. Liu, J. Ren, and Z. Guo, "Adaptive general scale interpolation based on weighted autoregressive models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 200–211, Feb 2015.

[11] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1382–1394, Apr 2013.

[12] Y. Romano, M. Protter, and M. Elad, "Single image interpolation via adaptive nonlocal sparsity-based modeling," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3085–3098, Jul 2014.

[13] J. J. Huang, W. C. Siu, and T. R. Liu, "Fast image interpolation via random forests," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3232–3245, Oct 2015.

[14] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec 2010.

[15] Forest Agostinelli, Michael R Anderson, and Honglak Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *Proc. Annual Conf. Neural Information Processing Systems*, pp. 1493–1501. 2013.

[16] Junyuan Xie, Linli Xu, and Enhong Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Annual Conf. Neural Information Processing Systems*, pp. 341–349. 2012.

[17] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[18] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. IEEE European Conf. Computer Vision*, pp. 184–199. 2014.

[19] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2016.

[20] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2016.

[21] Jie Ren, Jiaying Liu, Wei Bai, and Zongming Guo, "Similarity modulated block estimation for image interpolation," in *Proc. IEEE Int'l Conf. Image Processing*, 2011, pp. 1177–1180.

[22] W. Yang, J. Liu, M. Li, and Z. Guo, "General scale interpolation based on fine-grained isophote model with consistency constraint," in *Proc. IEEE Int'l Conf. Image Processing*, Oct 2014, pp. 1857–1861.

[23] M. Li, J. Liu, J. Ren, and Z. Guo, "Adaptive general scale interpolation based on weighted autoregressive models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 200–211, Feb 2015.

[24] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, Jun 2015, pp. 5197–5206.

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

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int'l Conf. Computer Vision*, Jul 2001, vol. 2, pp. 416–423.

[27] J. C. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.

[28] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Trans. Multimedia*, New York, 2014, pp. 675–678.