# DEEP PYRAMID VARIATION LEARNING FOR IMAGE INTERPOLATION

*Fu Qiang*[1], *Wenhan Yang*[2], *Ying Li*[1], *and Jiaying Liu*[2*]

[1] School of Software and Microelectronics, Peking University, Beijing, 100080
[2] Institute of Computer Science and Technology, Peking University, Beijing 100871

## ABSTRACT

Previous learning-based interpolation methods do not consider multi-scale structural information, which is generally effective for image modeling. In this paper, we design a deep network based on a novel pyramid variation learning approach with multi-scale structure modeling. An image is represented as multi-dimensional features. Besides two spatial dimensions, the features include a *neighboring variation* dimension where every pixel is encoded as the variation to its nearest low-resolution pixel, and a *scale* dimension along which the feature maps generated by a gradual down-sampling process are stacked. Thus, these multi-dimensional features are constructed to model local dependency and multi-scale similarity jointly. Inspired by this feature design, we build an end-to-end trainable **R**ecurrent **M**ulti-**P**ath **A**ggregation **N**etwork (**RMPAN**) for image interpolation, where the *scale* dimension is unfolded to form a multi-path aggregation network to apply joint filters at different scales recurrently. Location-aware sampling layers are used in RMPAN to transform feature maps into different scales with only location changes in each convolution path, which aggregate the context information without resolution loss. Comprehensive experiments demonstrate that our method leads to a superior performance and offers new state-of-the-art benchmark.

***Index Terms***— Deep learning, image interpolation, multi-scale similarity, pyramid structure, variation learning

## 1. INTRODUCTION

Image interpolation is a fundamental research topic in image processing, and thus effectively supports a wide range of applications, such as video surveillance and image/video display. The goal is to reconstruct a high-resolution (HR) image from one of its down-sampled low-resolution (LR) versions by inferring all missing pixels during the down-sampling process. Various interpolation methods could be classified into three groups: *polynomial-based methods*, *geometry-guided methods* and *learning-based methods*.

*Polynomial-based methods*, such as Bilinear and Bicubic methods [1], generate values of missing pixels by convolving neighboring pixels with fixed kernels. Their computational complexity is relatively low. However, their results include noticeable artifacts (*e.g.* blurring, ringing, jaggies and zippering) and unnatural representations of edges.

In order to utilize local structural information and obtain visually satisfactory results, *geometry guided methods* are proposed, categorized into explicit and implicit geometry guided methods. Explicit geometry guided methods [2] detect the geometric features, such as local covariance and edges, in an explicit manner and adjust the interpolation lattice based on structural directions dynamically. Implicit geometry guided methods [3–5] construct an optimization function with statistical geometric information and maximize this optimization function to obtain an adaptive filter for the missing pixels calculation in local regions. This soft modeling describes the intrinsic correlations of LR and HR pixels and achieves superior performance. However, the optimization function is built only based on information of a local region, effective knowledge of external images have not been explored. There still is room to further improve image interpolation.

Recently, the rapid development of deep neural networks also leads to the blooming of deep learning-based image processing, including denoising [6], super-resolution [7–10], video compression [11–13], rain removal [14–18]. Likewise, *learning-based methods* that acquire the mapping between LR pixels and missing HR ones from a paired training dataset achieve promising results with rather low computational complexity. In [19, 20], sparse dictionary learning, nonlocal patch prior and autoregressive model form an integrated optimization function to effectively make full use of both local dependency and nonlocal similarity. However, without using external information beyond the given LR image, this method only leads to a poor performance in non-repetitive regions. In [21], random forests project the natural image patch into different subspaces, and then to transform the LR patch into an HR one in the respective space. However, the subspace partition and local regression are optimized separately, and the regression model is a linear model, which limits its modeling capacity for complex mappings. In [22], Yang *et al.* made the first attempt to apply deep networks for image interpolation. In this network, a pixel is encoded as the summation of its near-

*Corresponding author. Email: liujiaying@pku.edu.cn. This work was supported by National Natural Science Foundation of China under contract No. 61772043, Beijing Natural Science Foundation under contract No. L182002 and No. 4192025, and CCF-DiDi BigData Joint Lab.

978-1-5386-9552-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICME.2019.00100

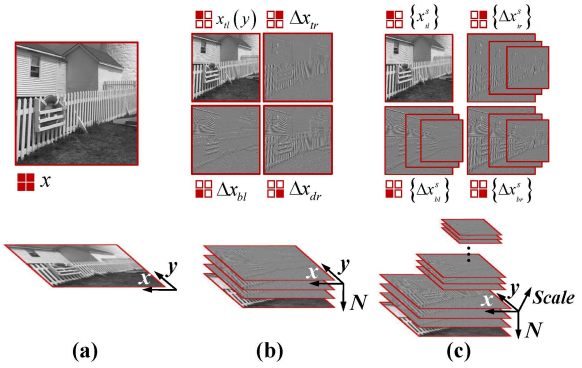



**Fig. 1**. The multi-dimensional pyramid variation representation is built based on the local redundancy and multi-scale similarity. (a) Original image representation. (b) Each pixel is encoded as the variation value. (c) A multi-dimensional pyramid is constructed by extending along the scale axis with gradual down-samplings.

est LR pixel and a learned variation. In this learning process, abundant structural correspondences in the pixel variation space are provided to facilitate inferring the lost information caused by image degradation. However, it does not explore a potential effective prior on image modeling – multi-scale structural information.

In this paper, we incorporate multi-scale structural information into the learning process and design a pyramid variation learning for image interpolation. An image is represented by multi-dimensional features. Besides spatial dimensions, there are two additional dimensions – A *neighboring variation* dimension where every pixel is encoded as the variation to its nearest LR pixel based on local similarity, and a *scale* dimension along which the feature maps generated by a gradual down-sampling process are stacked to model multi-scale similarities. With this feature structure, we unfold the *scale* dimension of the multi-dimensional features to build a **R**ecurrent **M**ulti-**P**ath **A**ggregation **N**etwork (**RMPAN**). We use location-aware sampling layers and convolutional layers to construct RMPAN. This location-aware sampling layer transforms feature maps into different scales with only location changes in each convolution path to aggregate context information without resolution loss. Experimental results demonstrate superiority of our RMPAN.

## 2. FROM VARIATION LEARNING TO PYRAMID VARIATIONAL LEARNING

In this section, we first review the variation learning briefly proposed in [22]. Then, considering multi-scale signal structure, we further propose our pyramid variation learning.

### 2.1. Variation Learning

The direct end-to-end learning from the given LR pixels to the missing HR ones suffers from three deficiencies. First, the low-frequency parts prevent the regression model from capturing the mapping relationship between low-frequency parts and high-frequency details. Second, the priors are imposed on the whole $x$ instead of high frequency image signal. Third, the structural correspondences of high frequency details are underneath.

Then, variation image representation is developed to get rid of the auto-correlation of $\mathbf{x}$ and $\mathbf{y}$ as well their correlation, and to make use of more useful structural correspondences within an image. Intuitively, as shown in Figs. 1 (a) and (b), Based on the local redundancy, a local pixel can be decomposed into one of its nearest neighbors and a small difference value, called the variation in [22] and our work. Correspondingly, an HR pixel could be decomposed into the top-left LR pixel in the corresponding $2 \times 2$ non-overlapping patch (for convenience, we use $2\times$ enlargement as example in our work but note that, our approach is general to apply for other times enlargement) and a difference value between the LR and HR pixels. This changes removes much of auto-correlation within the image. Formally, an HR image $\mathbf{x}$ is split into four parts $\mathbf{x}_{tl}$, $\Delta\mathbf{x}_{tr}$, $\Delta\mathbf{x}_{bl}$ and $\Delta\mathbf{x}_{dr}$.

$$\begin{aligned} \mathbf{x}_{tl} &= y, \\ \Delta\mathbf{x}_{tr} &= \mathbf{x}_{tr} - \mathbf{x}_{tl}, \\ \Delta\mathbf{x}_{bl} &= \mathbf{x}_{bl} - \mathbf{x}_{tl}, \\ \Delta\mathbf{x}_{br} &= \mathbf{x}_{bl} - \mathbf{x}_{tl}, \end{aligned} \tag{1}$$

We stack the last three terms as a tensor as shown in Fig. 1 (b) where two axises denote the locations, and another axis denotes the neighboring domain.

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

$$\mathbf{x} = \mathbf{x}_{\epsilon} + \Delta\mathbf{x}, \tag{2}$$

where $\mathbf{x}_{\epsilon}$ signifies 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 Eqn. (1) with $\Delta\mathbf{x}_{tl} = 0$. In image interpolation, $\mathbf{x}_{\epsilon}$ is given (equivalent to $\mathbf{y}$), thus we can aim to estimate $\Delta\mathbf{x}$ and leads to a new learning way:

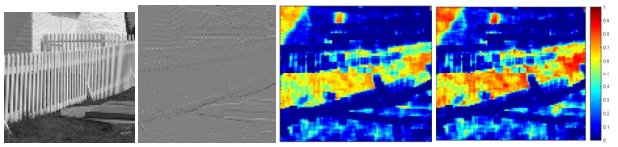
$$\mathbf{x} = f_r(\mathbf{y}) + \mathbf{x}_{\epsilon}. \tag{3}$$

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

### 2.2. Pyramid Variation Learning

The above-mentioned part goes through local dependency modeling, we then aim to model the multi-scale similarity. An image pyramid is built on a gradual down-sampling operation with a small factor. This repetitive operation constructs more similar patches because many typical structures are self-similar at different scales, which inspires us to extend previous 3D representation into a 4D one, with a new axis to signify the scale. Then, the feature representation in Fig. 1 (b) becomes Fig. 1 (c). And Eqn. (3) is further extended to a multi-scale form:

$$\mathbf{x} = \sum_{i=1}^{S} f_s^i(\mathbf{y}) + \mathbf{y}, \tag{4}$$



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

**Fig. 2**. Patch repetitiveness of a single image in two feature space. (a) Original image $\mathbf{x}_{tl}$. (b) Difference image $\Delta\mathbf{x}_{tr}$. (c)-(d) The heat maps for patch repetitiveness of (b) in the variation space in Eqn. (3) and the pyramid variation space in Eqn. (4), respectively. Red signifies high values, Blue signifies low values.

where $f_s^i(\cdot), i = 1, 2, ..., S$ are the learned inverse processes to estimate $\mathbf{x} - \mathbf{y}$ based on the whole multi-scale pyramid of $\mathbf{y}$. $S$ is the number of scales in our model.

Compared with Eqn. (3), the novel proposed image representation Eqn. (4) has more *structural correspondences*. From the perspective of structural correspondences on their patch repetitiveness, the potential redundancy within an image is measured. 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 (c) and (d) are the heat maps for the patch repetitiveness of (b) – the difference image – in the variation space in (3) and that in the pyramid variation space in (4), respectively. In these heat maps, the colors from red to blue signify the decrease of patch repetitiveness values. Compared with (c), regions in (d) significantly increase the patch repetitiveness. Among all representations in Fig. 2, our proposed pyramid variation model presents the most frequent patch repetitiveness.

## 3. RECURRENT MULTI-PATH AGGREGATION NETWORK FOR IMAGE INTERPOLATION

In this section, we turn the proposed pyramid variation learning into a novel recurrent multi-path aggregation network (RMPAN) for image interpolation.

### 3.1. The Overall Network Structure

The traditional convolutional neutral network (CNN) is capable to handle at most 3D dimension representation, thus we need to seek a solution to model our 4D representation. To address the problem, we unfold the scale axis to build a recurrent multi-path aggregation network (RMPAN). Each path aims to model the representation at a certain scale as shown in Fig. 3. To model the first three dimensions, our RMPAN takes a recurrent convolutional structure that performs a progressive signal recovery. The features are transformed and enhanced progressively. In each recurrence, the multi-path convolutions apply multi-scale filter operations on the image pyramid. For each path, we down-sample the feature maps by a certain scale via the proposed location-aware down-sampling, then perform two successive convolutions, and finally up-sample the feature maps via the location-aware up-sampling. With this coupled location-aware sampling operations (layers), we could effectively utilize a joint filter operation at a certain scale without resolution loss. The filtered results of multi-path convolutions are aggregated by a summation. After that, the variation map is reconstructed by the last convolution on the aggregated results. The proposed RMPAN 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 pixel lattice.

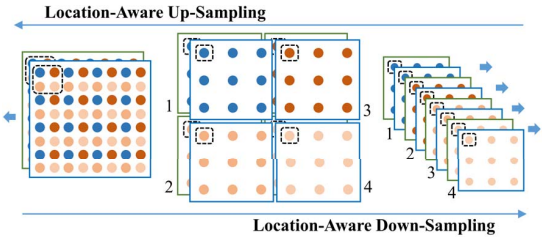



**Fig. 4**. Illustration for location-aware sampling layers. The up-sampling layer combines small feature maps into a larger one while the down-sampling layer splits a large feature map into several small ones.

### 3.2. Pyramid Variation Learning Network

Specifically, we illustrate each part in formulation:

**Feature extraction and reconstruction**. The first convolution extracts features $\mathbf{f}_{\text{in}}^1$ from the input LR image, and the penultimate convolution layer generates 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 given as follows,

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

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

where $\mathbf{y}$ denotes the input LR image, and $\mathbf{W}_{\text{extract}}$ and $\mathbf{b}_{\text{extract}}$ are the filter parameter and basis of the first convolution layer – feature extraction layer, respectively. As shown in Fig. 3, $\Delta\mathbf{x}_{\text{tl}}$, $\Delta\mathbf{x}_{\text{tr}}$,$\Delta\mathbf{x}_{\text{dl}}$ and $\Delta\mathbf{x}_{\text{dr}}$ denote the estimated top-left, top-right, down-left and down-right values in every $2 \times 2$ non-overlapping patch by the penultimate reconstruction layer. $\mathbf{W}_{\text{rect}}$ and $\mathbf{b}_{\text{rect}}$ are 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 combined with the LR image via addition to produce the corresponding HR pixels in these locations, and $\Delta\mathbf{x}_{\text{tl}}$ is equivalent to $\mathbf{y}$ based on the location correspondences in the degradation.

**Location-aware down-sampling and up-sampling.** To filter at different scales, a specialized network structure as shown in Fig. 4 is used to transform the feature map into different scales without loss of original information. For 2 times down-sampling, the features are divided into $2 \times 2$ non-overlapping

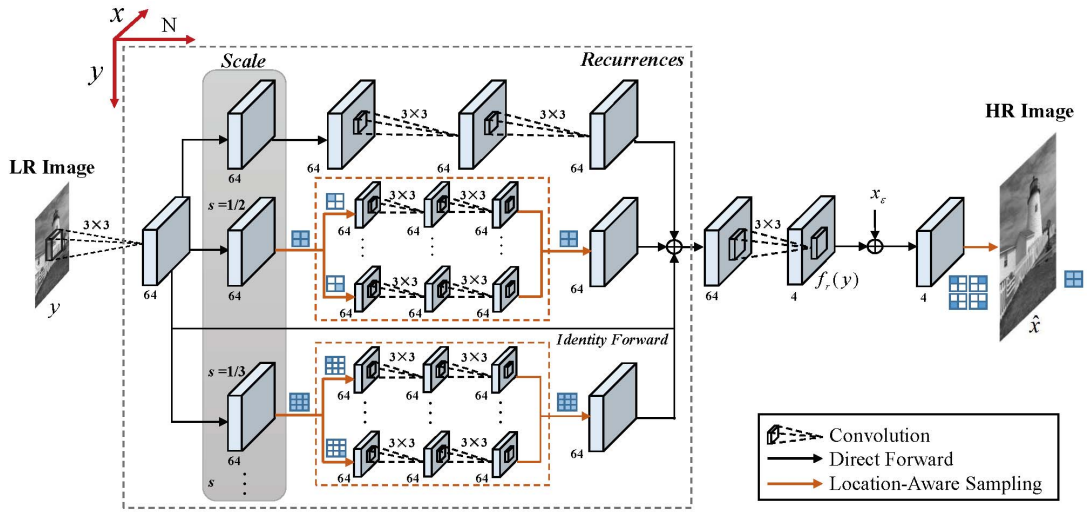



**Fig. 3**. The architecture of our proposed Recurrent Multi-Path Aggregation Network (RMPAN) that performs pyramid variation learning for image interpolation.

patches, and the four pixels in every patch are extracted into a feature map, respectively. Then, the four feature maps are stacked as a new feature map. For 2 times up-sampling, everything goes in a reverse way. The location-aware down-sampling layer works as

$$\mathbf{f}^k_{\text{spin},s,p}(\lfloor i \times s \rfloor, \lfloor j \times s \rfloor, c) = \mathbf{f}^k_{\text{in}}(i, j, c), \tag{7}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, $s$ signifies the scale of one convolution path, $p$ signifies the group of the output number. $i$ and $j$ denote the spatial location and $c$ denotes the channel number. 'spin' denotes the input features from the split process. $p$ is calculated as follows,

$$p = (i - \lfloor i \times s \rfloor - 1) \times 1/s + (j - \lfloor j \times s \rfloor) + 1. \tag{8}$$

Similarly, the location-aware up-sampling layer works as follows,

$$\mathbf{f}^k_{\text{concat},s}(i, j, c) = \mathbf{f}^k_{\text{spout},s,p}(\lfloor i \times s \rfloor, \lfloor j \times s \rfloor, c), \tag{9}$$

where 'spout' denotes the output features after processing the split results and 'concat' signifies that several feature maps are concatenated into one. By coupling the down-sampling and up-sampling layers with different $s$, the network could filter at different scales without sacrificing the resolution loss. For $s = 1$, we have

$$\mathbf{f}^k_{\text{spin},1,1}(i, j, c) = \mathbf{f}^k_{\text{in}}(i, j, c), \tag{10}$$

$$\mathbf{f}^k_{\text{concat},1}(i, j, c) = \mathbf{f}^k_{\text{spout},1,1}(i, j, c). \tag{11}$$

**Progressive feature enhancement**. Let $\mathbf{f}^k_{\text{in}}$ signify the input feature map at the $k$-th recurrence. The output feature map at the $k$-th recurrence, $\mathbf{f}^k_{\text{out}}$, is updated as follows,

$$\begin{aligned} \mathbf{f}^k_{\text{out}} &= \sum_s \max(0, M_s) + \mathbf{f}^k_{\text{in}}, \\ M_s &= \sum_p \mathbf{f}^k_{\text{concat},s,p}, \end{aligned} \tag{12}$$

$$\begin{aligned} \mathbf{f}^k_{\text{spout},s,p} &= \left(\mathbf{W}^k_{\text{spmid},s,p} * \mathbf{f}^k_{\text{spmid},s,p} + \mathbf{b}^k_{\text{spmid},s,p}\right), \\ \mathbf{f}^k_{\text{spmid},s,p} &= \max\left(0, \mathbf{W}^k_{\text{spin},s,p} * \mathbf{f}^k_{\text{spin},s,p} + \mathbf{b}^k_{\text{spin},s,p}\right), \end{aligned}$$

where $\mathbf{f}^k_{\text{in}} = \mathbf{f}^{k-1}_{\text{out}}$ denote the output features by the at $(k-1)$-th recurrence. $\mathbf{W}^k_{\text{spin},s,p}$ and $\mathbf{b}^k_{\text{spin},s,p}$ denote the filter parameter and basis of the first convolution in the $p$-th path at scale $s$ in the $k$-th iteration. $\mathbf{W}^k_{\text{spmid},s,p}$ and $\mathbf{b}^k_{\text{spmid},s,p}$ signify the filter parameter and basis of the second convolution in the $p$-th path at scale $s$ in the $k$-th iteration. The by-pass connection forwards $\mathbf{f}^k_{\text{in}}$ to $\mathbf{f}^k_{\text{out}}$. The feature map $\mathbf{f}^k_{\text{out}}$ can be regarded as the inferred $k$-th layer details of the feature maps.

**Network training.** Let $\mathbf{F}(\cdot)$ represent the learned network that recovers the HR image $\mathbf{x}$ based on the given LR image $\mathbf{y}$. We use $\mathbf{\Theta}$ to collectively signify all the parameters of the network as follows,

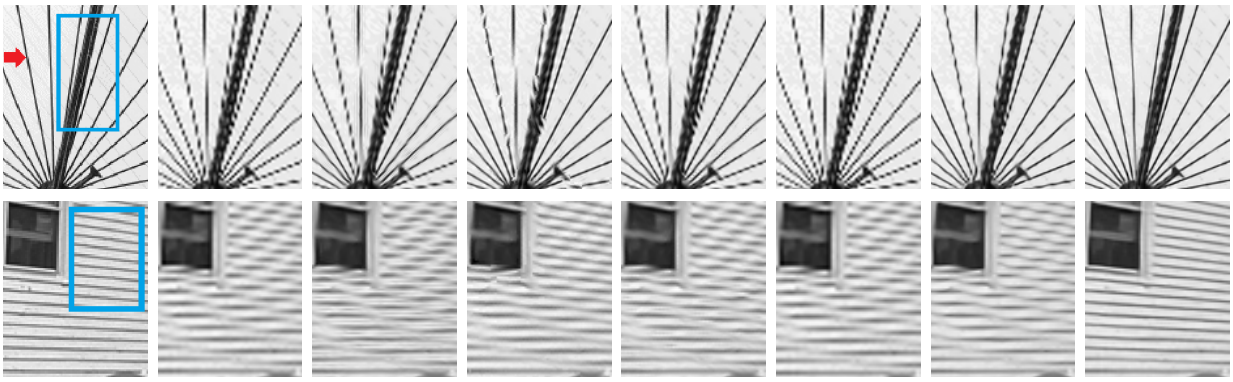
$$\begin{aligned} \mathbf{\Theta} = \{&\mathbf{W}_{\text{extract}}, \mathbf{b}_{\text{extract}}, \mathbf{W}_{\text{spin}}, \mathbf{b}_{\text{spin}}, \\ &\mathbf{W}_{\text{spmid}}, \mathbf{b}_{\text{spmid}}, \mathbf{W}_{\text{rect}}, \mathbf{b}_{\text{rect}}\}. \end{aligned} \tag{13}$$

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

$$L(\mathbf{\Theta}) = \frac{1}{n} \sum_{i=1}^{n} (||\mathbf{F}(\mathbf{y}_i, \mathbf{x}_i; \mathbf{\Theta}) - \mathbf{x}_i||^2. \tag{14}$$

## 4. EXPERIMENTAL RESULTS

**Experimental Setting**. The RMPAN is compared with conventional polynomial-based Bicubic interpolation method and seven state-of-the-art interpolation algorithms, including soft autoregressive interpolation (SAI) [4], similarity modulated block estimation (SMBE) [23], consistent segment adaptive gradient angle interpolation (CSAGA) [24], sparse



(a) HR (b) BC (c) SAI (d) AGSI (e) SMBE (f) NARM (g) FIRF (h) RMPAN

**Fig. 5**. Visual comparison between different algorithms. The top panel: *Bicycle* in *Set18*. The bottom panel: *Lighthouse* in *Set15*.

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

| Image | Bicubic | SAI | SMBE | CSAGA | SME | AGSI | NARM | FIRF | VLN | RMPAN |
|---|---|---|---|---|---|---|---|---|---|---|
| *Set15* | 28.81 | 29.19 | 29.32 | 29.20 | 29.26 | 29.06 | 29.47 | 29.81 | 30.23 | **30.55** |
| *Set18* | 28.82 | 29.44 | 29.47 | 29.29 | 29.35 | 29.33 | 29.75 | 30.11 | 30.63 | **30.84** |
| *Urban12* | 23.29 | 23.86 | 24.01 | 23.96 | 23.94 | 23.88 | 23.98 | 24.97 | 26.12 | **26.54** |

mixing estimators [25] (SME), adaptive general scale interpolation (AGSI) [5], nonlocal autoregressive modeling (NARM) [19], fast interpolation via random forest (FIRF) [21], and variational learning network (VLN) [22].

We compare our RMPAN 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 [21]. *Urban12* includes 12 urban landscapes images from *Urban* [26] dataset, that contains the images with many regular repetitive building patterns.

The input LR images are generated by down-sampling the original HR images with the scaling factor 2. Then, different interpolation methods are used to interpolate HR images from the input LR images. Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity index (SSIM) [27] are utilized as the criteria to evaluate the experimental results.

We trained our RMPAN with a training set including 591 images, consisting of 500 images in BSDS500 [28] and 91 images in [29]. We crop the images into $40 \times 40$ input and $80 \times 80$ output patches. Around 500,000 sub-images are generated by using a stride of 20 with the data augmentation of flipping and rotation. Our RMPAN was trained on Caffe platform [1] via stochastic gradient descent (SGD) with standard back-propagation. The momentum is set to 0.9, the initial learning rate is set to 0.001 for front-end layers and 0.00001 for the penultimate layer (before the fixed location up-sampling layer) during the training process. The learning rate is dropped by a factor of 10 when reaching 250,000 iterations. The batch size is set to 64. At most 300,000 back-propagations are allowed, which spent about 7 hours when training on a single Titan GTX 1080.

[1] http://caffe.berkeleyvision.org/

**Objective Evaluation**. The objective evaluation results are shown in Tables 1-2. The results clearly show that our method consistently outperforms other methods with significant performance gains. For *Set15*, our RMPAN achieves better performance than VLN with gains of 0.22dB and 0.0048 in PSNR and SSIM, respectively. For *Set18* and *Urban12*, larger performance gains are achieved, which are 0.21dB and 0.42dB in PSNR and 0.0031 and 0.0084 in SSIM, respectively.

**Subjective Evaluation**. We also present visual results of different methods in Fig. 5. The results clearly demonstrate the significant superiority of our RMPAN to other methods. It is observed that, Bicubic generates rather blurred results. Three AR-flavoured methods– SAI, AGSI and SMBE– improve the visual quality of their results significantly by increasing their adaptivity to model local image signals. However, artifacts and blurred results, such as the zigzag artifacts in the very thin line in the top panel of Fig. 5, are presented. NARM obtains better results on repetitive signal patterns because of its non-local similarity modeling. FIRF achieves very good visual quality, with the general dependency learned from a very large dataset. However, the intrinsic ill-posed nature of the problem caused by the image degradation also makes FIRF miscalculate HR signal, such as the wrong pattern prediction in the bottom panel of Fig. 5. By considering external general dependency and internal image pyramid priors, our RMPAN obtains better visual quality than other state-of-the-art methods. The superiority is obviously witnessed in regions of the axises in the top panel of Fig. 5, and the repetitive patterns of wall in the bottom panel of Fig. 5.

**Table 2**. The average SSIM Results on *Set15*, *Set18* and *Urban12*.

| Image | Bicubic | SAI | SMBE | CSAGA | SME | AGSI | NARM | FIRF | VLN | RMPAN |
|---|---|---|---|---|---|---|---|---|---|---|
| *Set15* | 0.8730 | 0.8784 | 0.8780 | 0.8782 | 0.8788 | 0.8743 | 0.8825 | 0.8831 | 0.8945 | **0.8993** |
| *Set18* | 0.8683 | 0.8765 | 0.8765 | 0.8757 | 0.8742 | 0.8733 | 0.8815 | 0.8829 | 0.8933 | **0.8964** |
| *Urban12* | 0.7977 | 0.8171 | 0.8192 | 0.8191 | 0.8141 | 0.8126 | 0.8235 | 0.8441 | 0.8797 | **0.8881** |

## 5. CONCLUSIONS

In this paper, we propose a novel multi-dimensional pyramid variation image representation and develop a recurrent multi-path aggregation network (RMPAN). 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, the pyramid variation learning and RMPAN are constructed for image interpolation. Making use of both local and nonlocal similarities jointly, the network applies joint filter operations recurrently, leading to a superior performance than previous methods and offering the new state-of-the-art.

## 6. REFERENCES


[1] R. Keys, "Cubic convolution interpolation for digital image processing," *TIP*, 1981.

[2] Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *TIP*, 2007.

[3] X. Li and M. Orchard, "New edge-directed interpolation," *TIP*, 2001.

[4] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *TIP*, 2008.

[5] M. Li, J. Liu, J. Ren, and Z. Guo, "Adaptive general scale interpolation based on weighted autoregressive models," *TCSVT*, 2015.

[6] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *TIP*, 2017.

[7] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *TPAMI*, 2016.

[8] W. Yang, S. Xia, J. Liu, and Z. Guo, "Reference guided deep super-resolution via manifold localized external compensation," *TCSVT*, 2018.

[9] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan, "Deep edge guided recurrent residual learning for image super-resolution," *TIP*, 2017.

[10] W. Yang, J. Feng, G. Xie, J. Liu, Z. Guo, and S. Yan, "Video super-resolution based on spatial-temporal recurrent residual networks," *CVIU*, 2018.

[11] Y. Hu, W. Yang, S. Xia, W. Cheng, and J. Liu, "Enhanced intra prediction with recurrent neural network in video coding," *DCC*, 2018.

[12] S. Xia, W. Yang, Y. Hu, S. Ma, and J. Liu, "A group variational transformation neural network for fractional interpolation of video coding," *DCC*, 2018.

[13] J. Liu, S. Xia, W. Yang, M. Li, and D. Liu, "One-for-all: Grouped variation network-based fractional interpolation in video coding," *TIP*, 2019.

[14] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *CVPR*, 2017.

[15] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *CVPR*, 2018.

[16] J. Liu, W. Yang, S. Yang, and Z. Guo, "D3r-net: Dynamic routing residue recurrent network for video rain removal," *TIP*, 2019.

[17] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or Fill? Deep Joint Recurrent Rain Removal and Reconstruction in Videos," *CVPR*, 2018.

[18] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, "Joint rain detection and removal from a single image with contextualized deep networks," *TPAMI*, 2019.

[19] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *TIP*, 2013.

[20] Y. Romano, M. Protter, and M. Elad, "Single image interpolation via adaptive nonlocal sparsity-based modeling," *TIP*, 2014.

[21] J. J. Huang, W. C. Siu, and T. R. Liu, "Fast image interpolation via random forests," *TIP*, 2015.

[22] W. Yang, J. Liu, S. Xia, and Z. Guo, "Variation learning guided convolutional network for image interpolation," *ICIP*, 2017.

[23] J. Ren, J. Liu, W. Bai, and Z. Guo, "Similarity modulated block estimation for image interpolation," *ICIP*, 2011.

[24] W. Yang, J. Liu, M. Li, and Z. Guo, "General scale interpolation based on fine-grained isophote model with consistency constraint," *ICIP*, 2014.

[25] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *TIP*, 2010.

[26] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," *CVPR*, 2015.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *TIP*, 2004.

[28] 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," *ICIP*, 2001.

[29] J. C. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *TIP*, 2010.