Practical Application of Random Forests for Super-Resolution Imaging

Jun-Jie Huang and Wan-Chi Siu
Centre for Signal Processing, Department of Electronic and Information Engineering
The Hong Kong Polytechnic University

***Abstract*—In this paper, a novel learning-based single image super-resolution method using random forest is proposed. Different from example-based super-resolution methods which search for similar image patches from an external database or the input image, and the sparse representation model based methods which rely on the sparse representation, this proposed super-resolution with random forest (SRRF) method takes the divide-and-conquer strategy. Random forest is applied to classify the training LR-HR patch pairs into a number of classes. Within every class, a simple linear regression model is used to model the relationship between the LR image patches and their corresponding HR image patches. Experimental results show that the proposed SRRF method can generate the state-of-the-art super-resolved images with near real-time performance.**

Keywords - Image processing, super-resolution; learning; random forest; fast approach.

I. INTRODUCTION

The objective of image super-resolution (SR) is to reconstruct a high resolution (HR) image from one or several low resolution (LR) images by up-sampling, deblurring and denoising. This problem is ill-posed, since there is not sufficient number of observations from the LR image, and the registration parameters are often unknown. SR techniques are useful and meaningful for many practical applications, for example, high-definition television, image coding, surveillance systems, medical imaging, etc.

The LR image in SR is assumed to be gotten from the HR image through blurring and down-sampling process, as

$$\mathbf{y} = \mathbf{DHx} + \mathbf{n}. \tag{1}$$

where $\mathbf{y}$ is the LR image which could be generated by the blurring operator $\mathbf{H}$ and down-sampling operator $\mathbf{D}$ from the original HR image $\mathbf{x}$ with additive noise $\mathbf{n}$.

In this paper, the SR setting is one of the three most typical SR scenarios. The degradation is realized by a bicubic filter followed by down-sampling by a scale of 2 (using MATLAB function *imresize* for down-sampling with the option "bicubic").

In the literature, there are mainly three classes of single image SR methods: (i) interpolation-based methods, (ii) reconstruction-based methods and (iii) learning-based methods. Interpolation-based methods [1-3] use non-adaptive or adaptive linear filters for up-sampling. They can realize fast SR, but produce blurry edges. Reconstruction-based methods [4-7] generate SR images by imposing certain prior knowledge, including, gradient-profile prior [4], nonlocal similarity [5] [6] and total-variation prior [7], etc. Learning-based methods [8-13] restore the HR image through a set of LR-HR training images. Among the learning-based methods, the example-based methods [8-10] search similar patches from an external database [8], from the input LR image itself across different scales [9] or from the local self-similar examples [10]; the sparse representation model based methods [11-14] use two coupled over-completed dictionaries and assume the desired HR image patch can be reconstructed using the same or similar sparse signal as that of the observed LR image patch and the coupled HR dictionary.

Random forest is an ensemble learning method and can achieve classification and regression. Various computer vision tasks [15-17] have applied random forest for real-time applications.

In this work, we propose a learning-based SR method using random forest (SRRF). Due to the complex relationship between LR image patches and HR image patches, a simple linear regression method cannot solve the SR problem. The basic idea of our work is to apply random forest for simultaneous classification and regression to the training LR-HR image patch pairs. The SR problem can thus be overcome through a combination of classification and simple linear regression. One of the drawbacks of most existing SR approaches is their high computational complexity which makes these approaches incapable for real-time applications, while our proposed SR using random forest method can give results in near real-time.

This paper is organized as follows. In Section II, the proposed SR using random forest method is introduced. In Section III, we present the experimental results and Section IV concludes the paper.

II. SUPER-RESOLUTION USING RANDOM FOREST

A. Super-Resolution Random Forest Training

The training data used in our paper is in the form of LR-HR patch pairs.

$$P_i = (\mathbf{L}_i, \mathbf{H}_i), \tag{2}$$

where $\mathbf{L}_i \in R^d$ is an image patch sampled from the initially upsampled LR image using bicubic interpolation, $\mathbf{H}_i \in R^d$ is the corresponding image patch of $\mathbf{L}_i$ and sampled from the original HR image and $\sqrt{d}$ is the patch size.

The objective of image super-resolution is to find a mapping function f^* which can minimize the mean squared error (MSE) between the mapped LR image patches and the original HR image patches:

$$f^* = \arg\min_f \frac{1}{N}\sum_{i=1}^{N} \| f(\mathbf{L}_i) - \mathbf{H}_i \|^2. \quad (3)$$

The initial bicubic interpolation can handle the smooth areas both efficiently and effectively. Hence, it is not necessary to find a mapping function for smooth areas. Only the patches containing the edge areas are further processed using the mapping function. The edge areas are determined on the initial bicubic up-sampled image using the Canny edge detector. If a pixel's edge magnitude of the Canny edge result is larger than a threshold (20), this pixel will be regarded as edge area.

Since a single function cannot adapt to varying appearances of all patches, the random forest is applied for simultaneously patch classification and regression. Random forest consists of a set of decision trees. Each decision tree is trained using a subset of the whole training data. A decision tree has non-leaf nodes and leaf nodes. Let us define the N training data at a node j as $S_j = \{P_i \mid i = 1, \ldots, N\}$. At each node of a decision tree, a mapping function will be learned according to (3). The mappling function is modeled by a linear regression model $\mathbf{C}$.

$$f(\mathbf{X}) = \mathbf{C}\mathbf{X}, \quad (4)$$

where $\mathbf{X} \in R^d$ and $\mathbf{C} \in R^d \times R^d$.

A closed form solution of (3) and (4) can be obtained by the least squares method:

$$\mathbf{C} = \mathbf{H}\mathbf{L}^T(\mathbf{L}\mathbf{L}^T)^{-1}. \quad (5)$$

With the learned linear regression model $\mathbf{C}$, the predicated HR image patch $\mathbf{H}^R$ can be reconstructed using the LR image patch $\mathbf{L}$:

$$\mathbf{H}^R = f(\mathbf{L}) = \mathbf{C}\mathbf{L}. \quad (6)$$

An entropy $H(S)$ formulation evaluates the goodness of the learned mapping function. The smaller entropy is, the better the learned mapping function is, and vice versa.

$$H(S) = \frac{1}{|S|}\sum_{i=1}^{|S|} \| \mathbf{H}^R - \mathbf{H} \|^2. \quad (7)$$

A non-leaf node j classifies the training data reached at this node S_j to its left child node or right child node; according to the split function $h(\mathbf{x}, \theta_j)$ with the best binary test parameter θ_j. The binary test parameter $\theta = (p, q, \tau)$ is a testing condition for the observed LR image patches and defined by two positions p, q on the LR image patch with a threshold value τ. The split function $h(\mathbf{x}, \theta)$ tests each training LR image patch belonging to S_j with condition θ to decide partitioning a training data P_i to the left child node S_j^L or the right child node S_j^R.

$$h(\mathbf{x}, \theta) = \begin{cases} 0, & \text{if } \mathbf{x}(p) < \mathbf{x}(q) + \tau \\ 1, & \text{otherwise} \end{cases}, \quad (8)$$

where $\mathbf{x}$ is the observed LR image patch.

$$\begin{aligned} S_j^L &= \{P_i \in S_j, P_i = (\mathbf{L}_i, \mathbf{H}_i) \mid h(\mathbf{L}_i, \theta) = 0\} \\ S_j^R &= S_j \setminus S_j^L \end{aligned}. \quad (9)$$

The information gain $I(S, \theta)$ is used to evaluate a binary split with the binary test parameter θ for the training data S_j. A large information gain means that the binary split divides the original training data at the parent node into two child nodes which can more effectively model the relationship between the LR image patches and the original HR patches.

$$I(S_j, \theta) = H(S_j) - \sum_{n \in \{L,R\}} \frac{|S_j^n|}{|S_j|} H(S_j^n). \quad (10)$$

At node j, a set of K randomly generated candidate binary test parameters is selected to try to split the training data into two child nodes. In extremely randomized trees [18] and for regression problem, the number of randomly generated binary test parameters should be:

$$K = d(d-1), \quad (11)$$

where d is the dimension of the feature vector.

The information gain of each binary test parameter is applied to assess that binary split. Among these binary test parameters, the binary test parameter which gives the highest information gain and its information gain is larger than 0is selected as the best binary test parameter θ_j.

$$\theta_j = \arg\max_\theta (I(S, \theta)). \quad (12)$$

A node will stop partitioning the training data in three circumstances: (i) the node reaches the pre-defined maximum tree depth D_{max}, (ii) the maximum information gain is smaller than 0, (iii) the number of training data is too few (less than the minimum number of training data N^T) to further split. If a node can be further spitted, this node is regarded as a non-leaf node; otherwise this node is treated as a leaf node. During training, each non-leaf node stores the best binary test parameter and each leaf node stores the linear regression model.

However, a totally randomly generated binary test parameter tends to split the training data into two unbalanced child nodes, i.e. the training data in two child nodes have great difference. We propose to apply a constraint to select the randomly generated binary test parameter. The relative size of two child nodes should be within a range as shown in (13). If a randomly generated binary test parameter cannot fulfill (13), this binary test parameter will be rejected. We are going to find K binary test parameters meeting (13) for each node as candidate binary test parameters. The constraint parameter λ is selected as 0.75 using cross-validation.

$$\max(|S^L|, |S^R|) \times \lambda \le \min(|S^L|, |S^R|). \quad (13)$$

B. Image Super-Resolution with Random Forest

During up-sampling, the input LR image has to be initially upsampled by bicubic interpolation to the desired size of the HR image. As stated in II. A, only patch pairs from edge areas will be included for training. Similarly, just the patches containing edge areas will be extracted for further processing.

TABLE II. PSNR(DB) OF THE PROPOSED METHOD SRRF(K) WITH\WITHOUT BACKPROJECTION WITH K=1,…,5.

	Without back-projection					*With back-projection*				
Images	SRRF(1)	SRRF(2)	SRRF(3)	SRRF(4)	SRRF(5)	SRRF(1)+BP	SRRF(2)+BP	SRRF(3)+BP	SRRF(4)+BP	SRRF(5)+BP
Baby	38.11	38.14	38.14	38.15	38.15	38.44	38.46	38.47	38.48	38.48
Bird	39.94	40.14	40.16	40.15	40.15	40.42	40.61	40.64	40.64	40.63
Butterfly	31.79	31.85	31.99	32.06	32.08	32.15	32.20	32.34	32.41	32.43
Head	35.49	35.50	35.50	35.50	35.50	35.71	35.72	35.72	35.73	35.72
Woman	34.93	34.84	34.91	34.98	34.99	35.24	35.14	35.21	35.28	35.28
Average	36.05	36.09	36.14	36.17	36.17	36.39	36.43	36.47	36.51	36.51

TABLE I. PSNR(DB) AND COMPUTATIONAL TIME (S) COMPARISON.

	Bicubic		ScSR [11]		Zeyde [12]+BP		ANR [13]+BP		BPJDL [14]		**SRRF(1)+BP**		**SRRF(4)+BP**	
Images	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
Baby	37.10	0.00	26.10	86.9	38.39	5.7	38.48	1.6	38.52	1274.9	38.44	0.26	38.48	0.64
Bird	36.80	0.00	27.30	31.2	40.15	1.7	40.12	0.5	40.63	352.2	40.42	0.10	40.64	0.27
Butterfly	27.40	0.00	23.00	32.8	30.90	1.4	30.64	0.4	31.44	218.0	32.15	0.09	32.41	0.21
Head	34.90	0.00	27.40	25.3	35.68	1.7	35.69	0.4	35.74	363.6	35.71	0.09	35.73	0.23
Woman	32.10	0.00	26.40	29.5	34.70	1.6	34.65	0.5	35.25	323.0	35.24	0.09	35.28	0.21
Average	33.66	0.00	26.04	41.14	35.96	2.43	35.92	0.68	36.32	506.3	36.39	0.13	36.51	0.31

For an extracted patch $\mathbf{x} \in R^d$, it is passed to every decision tree in the random forest. Within each decision tree, $\mathbf{x}$ is recursively partitioned to left or right child node according to the split function result with the binary test parameters stored at the non-leaf nodes, until it reaches a leaf node. An associated linear regression model at the reached leaf node will be applied for HR image patch prediction. Since there are multiple decision trees in the random forest, the input LR image patch $\mathbf{x}$ possesses multiple linear regression models. These linear regression models will be averaged to form a final linear regression model for prediction.

$$\mathbf{y} = (\frac{1}{N}\sum_{i=1}^{N}\mathbf{C}_i)\mathbf{x}. \tag{14}$$

where N is the number of decision trees in the random forest, $\mathbf{C}_i$ is the linear regression model obtained in the i^{th} decision tree, $\mathbf{x}$ is the LR image patch and $\mathbf{y}$ is the predicted HR image patch.

The random forest can provide better quality than single decision tree. The linear regression models from different decision trees all generate some errors with respect to the original signal, but the errors are de-correlated. By combining models from the decision trees, the errors are suppressed with the fused linear regression model. With more decision trees (more computational time) in the random forest, the quality of the super-resolved image becomes finer. Thus, the proposed method is a scalable image super-resolution method.

III. EXPERIMENTAL RESULTS

To evaluate the proposed image super-resolution method, 5 commonly used images are adopted for testing as shown in Fig. 1. For color image, only the luminance channel (Y in the YCbCr color space) will be considered for objective comparison.

We compare our proposed method with the state-of-the-art image super-resolution methods including: the bicubic interpolation method, the super-resolution via sparse coding (ScSR) method [11], the Zeyde's K-SVD based sparse coding method [12], the anchored neighborhood regression (ANR) method [13], and the Beta process joint dictionary learning (BPJDL) method [14]. The PSNR is applied for objective comparison.

Fig.1. 5 testing images. From left to right: *Baby, Bird, Butterfly, head* and *Woman.*

In this paper, we only evaluate the up-sampling results with a factor of 2. The patch size $\sqrt{d}$ was selected as 5, the maximum tree depth D_{max} is 12, the minimum number of training data for further split N^T is 200, and the constraint parameter λ is 0.75. There are 5 decision trees in the super-resolution random forest. Each decision tree samples around 1200000 LR-HR patch pairs from natural images (as in [11]) for training and has around 3800 leaf nodes. We apply back-projection (BP) to the results of our proposed SRRF method, Zeyde method and ANR method for refinement to make the comparison fair, since the ScSR method and the BPJDL method have employed back-projection.

Let us define SRRF(k) as the proposed SRRF method with k decision trees. TABLE I shows the PSNR of the proposed SRRF(k) method with\without back-projection with k=1,…,5. From the experimental results, the PSNR of the proposed method improves quickly when the decision tree number increases from 1 to 4, and tends to go steady with the 5$^{\text{th}}$ decision tree. With back-projection, the PSNR is around 0.3dB higher compared without back-projection [19].

TABLE II compares the proposed SRRF(1)+BP, SRRF(4)+BP method with the other methods with respect to PSNR and computational time. Our proposed method is implemented in C++, while all other methods were implemented in Matlab. We assume C++ implementation is 10 times faster than Matlab implementation and magnify the computational time of the proposed method 10 times for comparison. We can find that the proposed SRRF method

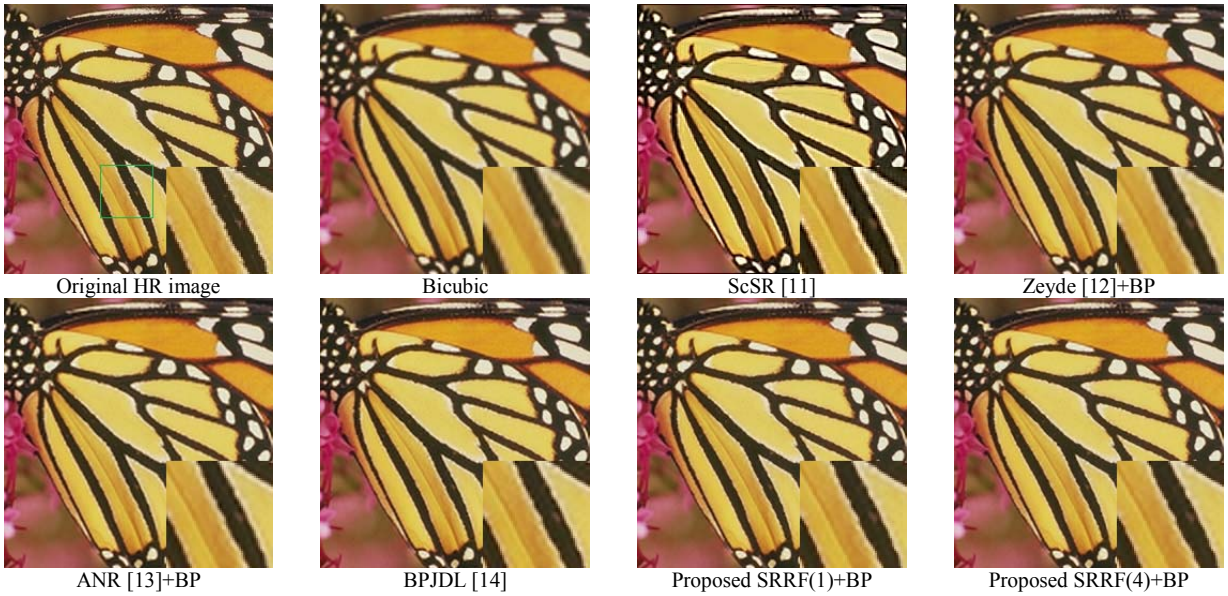

Fig.2. Subjective comparison between our proposed SRRF(1)+BP, SRRF(4)+BP method and Bicubic, ScSR method , Zeyde+BP method, ANR+BP method and BPJDL method.

requires comparable computational time as Zyede [12] and ANR [13], while our method achieves 0.5~0.6dB improvement in PSNR. The BPJDL method can generate close results as our proposed method, but it requires around 400 times computational time. Fig.2 provides the subjective comparison between our proposed SRRF(1)+BP, SRRF(4)+BP and Bicubic, ScSR method , Zeyde+BP method, ANR+BP method and BPJDL method. Our approach generates sharper edges.

IV. CONCLUSIONS

In this paper, we have proposed a novel image super-resolution method with random forest (SRRF). The proposed SRRF method demonstrated its superiorities comparing with the state-of-the-art image super-resolution methods. It achieves the highest PSNR among other methods, while the required computational cost is only around 0.5% with respect to the second best method BPJDL. Since the SRRF has very little computation load and the random forest is suitable for parallel programming, this new approach is a suitable candidate for image super-resolution in practical applications. For future work, one could investigate a more compact random forest structure for image super-resolution to further suppress the computational cost.

ACKNOWLEDGMENT :

This work was supported in part by the Center for Signal Processing, the Hong Kong Polytechnic University, and the Research Grant Council of the Hong Kong SAR Government under B-Q38S.

REFERENCES

[1] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process*., vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

[2] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Process*., vol. 26, no. 6, pp. 508–517, Dec. 1978.

[3] K.W. Hung and W.C. Siu, "Robust Soft-decision Interpolation using weighted Least Squares," *IEEE Trans. Image Process*., vol.21, no.3, pp.1061-1069, March 2012, USA.

[4] J. Sun, Z. Xu, and H. Y. Shum. Image super-resolution using gradient profile prior. In CVPR, 2008.

[5] H. Zhang, J. Yang, Y. Zhang, and T. Huang. Non-local kernel regression for image and video restoration. In ECCV, 2010.

[6] K.W. Hung and W.C. Siu, "Single-Image Super-Resolution Using Iterative Wiener Filter," Proc. IEEE Int. Conf. Acoustics, Speech, Sig. Pro. (ICASSP-2012), pp. 1269-72, 25-30, Mar., 2012, Kyoto Japan.

[7] A. Marquina and S. J. Osher, "Image super-resolution by TV regularization and Bregman iteration," *J. Sci. Comput.*, vol. 37, no. 3, pp. 367–382, 2008.

[8] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based superresolution," *IEEE Trans. Comp. Graph. App*., vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.

[9] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," Proceedings, pp. 349–356, *2009 IEEE International Conference on Computer Vision (ICCV'09)*, 29 Sep-2 Oct, 2009, Kyoto, Japan.

[10] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph*., vol. 30, no. 1, pp. 1-10, 2010.

[11] J. Yang, J. Wright, T. Huang, and Y. Ma. Image superresolution via sparse representation. IEEE Trans. on Image Processing, 19(11), 2010.

[12] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representation. In Proc. of Internation Conference on Curves and Surfaces, 2010.

[13] R. Timofte, V. De Smet, L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution", In Proc. of ICCV, pp. 1920–1927, 2013.

[14] L. He, H. Qi, and R. Zaretzki, "Beta process joint dictionary learning for coupled feature spaces with application to single image superresolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 345–352.

[15] J. Gall, A. Yao, N. Razavi, L. Gool, and V. Lempitsky, "Hough forests for object detection, tracking, andaction recognition," *IEEE Trans. Pattern Anal. Mach. Intel*., vol. 33, no. 11, pp. 2188–2202, Nov. 2011.

[16] V. Lepetit, and P. Fua, "Keypoint recognition using randomized trees," *IEEE Trans. Pattern Anal. Mach. Intel*., vol. 28, no. 9, pp. 1465-1479, Sept. 2006.

[17] P. Dollár and C. L. Zitnick, "Structured Forests for Fast Edge Detection," Proceedings, *2013 IEEE International Conference on Computer Vision,* pp. 1841-1848, 1-8 Dec. 2013, Sydney, Australia.

[18] P. Geurts, D. Ernst, and L.Wehenkel, "Extremely randomized trees," *Machine Learning*, 63.1 (2006):3–42.

[19] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion and transparency," *J. Vis. Commun. Image Represent.*, vol. 4, no. 4, pp. 324–335, 1993.