IMAGE SUPER-RESOLUTION USING MULTI-LAYER SUPPORT VECTOR REGRESSION

Jie Xu†, *Cheng Deng*†, *Xinbo Gao*†, *Dacheng Tao*‡, *and Xuelong Li*††

†School of Electronic Engineering, Xidian University, Xi'an 710071, China
‡Centre for Quantum Computation & Intelligent Systems, University of Technology, Sydney, Australia
††Xi'an Institute of Optics and Precision Mechanics, CAS, Xi'an 710119, China

ABSTRACT

Existing support vector regression (SVR) based image super-resolution (SR) methods always utilize single layer SVR model to reconstruct source image, which are incapable of restoring the details and reduce the reconstruction quality. In this paper, we present a novel image SR approach, where a multi-layer SVR model is adopted to describe the relationship between the low resolution (LR) image patches and the corresponding high resolution (HR) ones. Besides, considering the diverse content in the image, we introduce pixel-wise classification to divide pixels into different classes, such as horizontal edges, vertical edges and smooth areas, which is more conductive to highlight the local characteristics of the image. Moreover, the input elements to each SVR model are weighted respectively according to their corresponding output pixel's space positions in the HR image. Experimental results show that, compared with several other learning-based SR algorithms, our method gains high-quality performance.

***Index Terms*—** Super-resolution (SR), support vector regression (SVR), multilayer, pixel-wise classification

1. INTRODUCTION

Image super-resolution (SR) technique is to simulate the process of image degradation of imaging system (optical blur, motion blur, undersampling and system noise, etc), and to generate a raster image with a higher resolution than its source [1]. The existing SR methods can be roughly divided into three categories: interpolation-based, reconstruction-based, and learning-based. Recently, learning-based SR methods have attracted ever-increasingly attention due to its ability to restore high frequency details in texture areas with the help of other example images. The essence of these methods is to establish a model between low-resolution (LR) image and its corresponding high-resolution (HR) one, and then to estimate the model parameters using the training set. However, these methods suffer from serious boundary effect, for the reason that they only construct the mapping relationship from LR patches to HR patches.

To solve the boundary effect occurred in learning-based SR techniques, some investigations on the application of SVR to SR problem are conducted from different perspectives. Ni et al. [2] proposed an SVR-based SR method, which takes the LR image patches removed the center pixel as input and the corresponding HR image patches as output, respectively. Unfortunately, this method ignored the significant role of the center pixel in the LR image patch. Along this way, Li et al. [3] adopted the interpolated image patches as input and the center pixel of corresponding error image (i.e., the difference between the original image and the interpolated one) patches as output. Although this method attempted to build a mapping between patch and pixel instead of between patch and patch to overcoming boundary effect, it makes the resulting image over-smoothing.

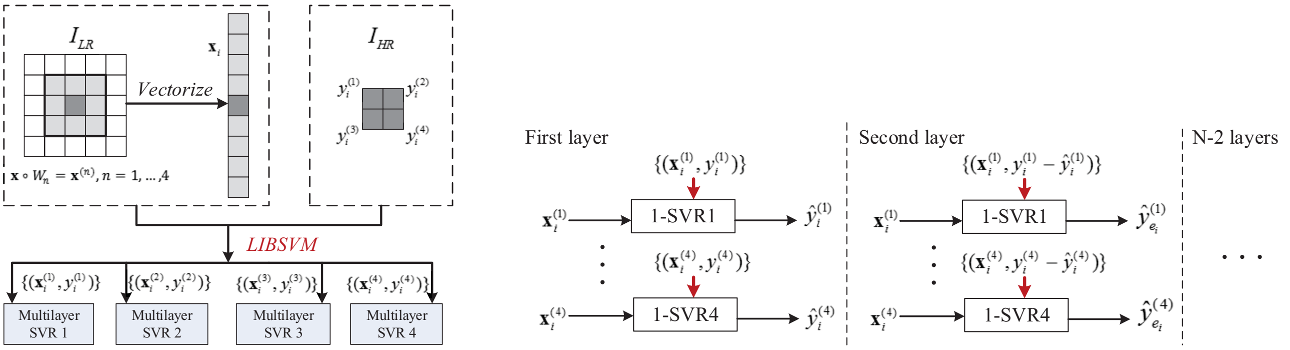
In this paper, we propose a multi-layer SVR based SR algorithm to address the aforementioned issue. We use SVR to model the relationships not only between LR image to its corresponding HR one, but also between LR image and the error image (i.e., the difference between pseudo high-frequency image and the original image), by extending single layer SVR model to a multi-layer model. Due to the fact that training SVR model in the whole image directly may not consider the diverse content in the image, we introduce pixel-wise classification to divide pixels into horizontal edges, vertical edges and smooth areas for training. Moreover, we weight the input set of each SVR model according to the space position of its corresponding pixels in the HR image patches, respectively.

The remainder of this paper is organized as follows. Section 2 introduces support vector regression. Then we describe our algorithm and analyze our model for imaging systems in Section 3. Experimental results are given in Section 4. Section 5 concludes this paper.

2. SUPPORT VECTOR REGRESSION

Support vector regression (SVR) machine is an application of support vector in the field of regression function [4]. The SVR classification is that the sample points only have one category which seeks the optimal hyperplane not to make two types of sample points share the "maximum margin", but to make all the sample points smallest away from the "total deviation". While the sample points are between two boundaries,

978-1-4799-2893-4/14/$31.00 ©2014 IEEE



(a) (b)

Fig. 1. Multi-layer SVR algorithm: (a) procedure for SVR super-resolution; (b) explanation of multi-layer SVR.

seeking the optimal regression hyperplane equals to find the maximum margin [5].

2.1. SVR Model

In linear case, SVM fitting function first considers using a linear regression function $f(x) = \omega \cdot x + b$ to fit (x_i, y_i), $i = 1, 2, ..., n$, where $x_i \in \mathbb{R}^n$ is input and $y_i \in \mathbb{R}$ is output. That is, we need to determine ω and b.

Penalty function is a measure of error in model learning process, generally selected before learning. Different learning problems correspond to different loss functions. Standard support vector machine adopts ϵ-nonsensitivity function, that is, with an assumption that all the training data are fitted by a linear function under ϵ-accuracy

$$\begin{cases} y_i - f(x_i) \leq \epsilon + \xi_i \\ f(x_i) - y_i \leq \epsilon + \xi_i{}^* \quad i = 1, 2, ..., n \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{1}$$

where ξ_i, ξ_i^* are relaxation factors.

In this case, the problem is transformed into an object function minimization problem

$$R(\omega, \xi, \xi^*) = \frac{1}{2}\omega \cdot \omega + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{2}$$

where the first term is to make the fitting function smoother and to enhance the generalization capability; the second term is to reduce error; and the constant C expresses the degree of punishment for the samples that exceed the error ϵ.

2.2. ϵ-Support Vector Regression (ϵ-SVR)

The basic idea of a nonlinear SVR model is to map the input vectors into a high-dimensional feature space (Hilbert space) using a predetermined nonlinear mapping, and to do linear regression in the space.

First we map the input x into the high-dimensional feature space $\mathbb{H}$ through the mapping $\phi : \mathbb{R}^n \to \mathbb{H}$, and adopt $f(x) = \omega \cdot \phi(x) + b$ to fit data $(x_i, y_i), i = 1, 2, ..., n$. Consider a set of training points, $\{(x_1, z_1), , (x_l, z_l)\}$, where $x_i \in \mathbb{R}^n$ is a feature vector and $z_i \in \mathbb{R}^l$ is the target output. For parameters $C > 0$ and $\epsilon > 0$, the standard form of support vector regression [6] is

$$\begin{aligned} \min_{\omega, b, \xi, \xi^*} \quad & \frac{1}{2}w^\top w + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^* \\ \text{subject to} \quad & w^\top \phi(x_i) + b - z_i \leq \epsilon + \xi_i, \\ & z_i - w^\top \phi(x_i) - b \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, i = 1, ..., l. \end{aligned} \tag{3}$$

The dual problem is

$$\begin{aligned} \min_{\alpha, \alpha^*} \quad & \frac{1}{2}(\alpha - \alpha^*)^\top Q(\alpha - \alpha^*) \\ & + \epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} z_i(\alpha_i - \alpha_i^*) \\ \text{subject to} \quad & e^\top(\alpha - \alpha^*) = 0, \\ & 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, ..., l, \end{aligned} \tag{4}$$

where $Q_{ij} = K(x_i, x_j) \equiv \phi(x_i)^\top \phi(x_j)$. ϕ is unknown and high-dimensional. Support vector machine theories only consider the dot product $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ called kernel function in high-dimensional feature space and do not use function ϕ directly.

After solving the problem above, the approximate function is

$$\sum_{i=1}^{l}(-\alpha_i + \alpha_i^*)K(x_i, x) + b. \tag{5}$$

In LIBSVM, we output $(\alpha - \alpha^*)$ in the model.

3. SVR BASED SUPER-RESOLUTION

With the support vector regression and its solving method, we discuss how to apply SVR to image super-resolution in this section.

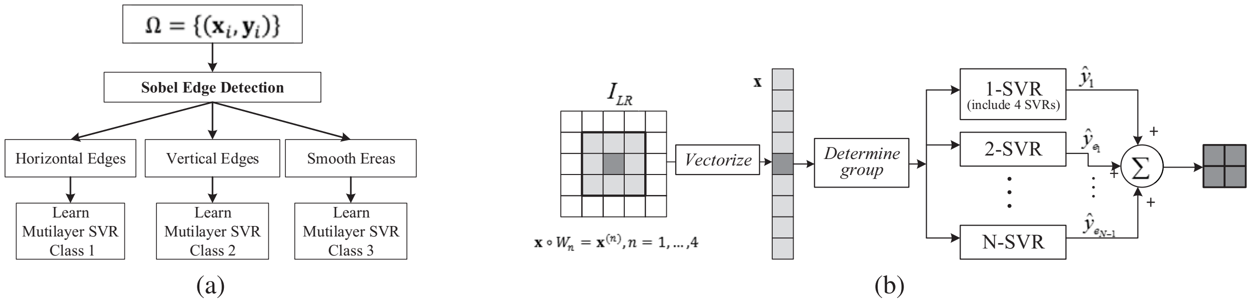


Fig. 2. Multi-layer SVR algorithm: (a) training algorithm; (b) testing algorithm.

3.1. General SVR Super-Resolution

Considering the diverse content in a whole image, we introduce pixel-wise classification to divide pixels into different classes, such as horizontal edges, vertical edges and smooth areas for training. Then, we do the same in each part. Given LR and HR image patches I_{LR} and I_{HR} with sizes $D \times D$ and $U \times U$, respectively, to super-resolve the center pixel of I_{LR} by a factor of U, we define vectors $\mathbf{x}$ and $\mathbf{y}$ in a given training set Ω with x_i features and y_i label pairs as follows,

$$\begin{aligned} \mathbf{x} &= W \circ \text{vectorize}\,(I_{LR}) \in \mathbb{R}^{D^2 \times 1} \\ \mathbf{y} &= \text{vectorize}\,(I_{HR}) \in \mathbb{R}^{U^2 \times 1} \end{aligned} \tag{6}$$

where W is a weighed matrix changing with different position of each pixel in an HR image patch. The task at hand is to super-resolve the center pixel of the patch with size $D \times D$ by a factor of U, and to predict U high resolution pixels corresponding to the center pixel. For instance, in the case of $2\times$ super-resolution, as shown in Fig. 1, $D = 5$ and $U = 2$.

This is a multi-output regression problem [7], which can be solved directly or be divided into several separate single output regression problems according to the current literature [8]. Here we adopt the second method, treating the multi-output regression problem as several separate single output regressions. Therefore, learning the four outputs becomes $\{y^{(j)} = g^{(j)}(x)\} \subset \mathbb{R}$ for $j = 1, ..., 4$, given the input $x \in \mathbb{R}^{N^2}$, and $g^{(j)}$ is estimated by ϵ-SVR in Eq. (4). To further improve the results, a multi-layer SVR model is proposed.

3.2. Multi-layer SVR Model

Suppose that the reconstructed image using one layer of SVR model is $\hat{I}_1$, the original HR image I_h can be expressed as

$$I_h = \hat{I}_1 + I_e \tag{7}$$

where the error image I_e is the difference between the HR image and the pseudo-high frequency image $\hat{I}_1$.

Introducing the second level SVR model is to reconstruct the error image $\hat{I}_{e_1}$, and to make it as close as possible to I_e,

$$I_e = \hat{I}_{e_1} + I_{e_2} \tag{8}$$

where I_{e_2} is the difference between I_e and $\hat{I}_{e_1}$.

Introducing the third level SVR model is to reconstruct the error image $\hat{I}_{e_2}$, and to make it as close as possible to I_{e_2}, as shown in Fig.1. By analogy, if we adopt N layers SVR model, we can obtain the final constructed results as follows:

$$I_h = \hat{I}_1 + \hat{I}_{e_1} + \hat{I}_{e_2} + ... + \hat{I}_{e_{N-1}} \tag{9}$$

4. EXPERIMENTS

In experimental parameter settings, we utilize LIBSVM [4], a software package for SVM pattern recognition and regression, to train SVR models. We empirically choose Radial Basis Function (RBF) as the kernel function, setting the parameter $g = 5$ and $c = 1$, and choose ϵ-SVR model. W is empirically selected according to the relative position of the pixels. Considering the tradeoff between computation cost and reconstruction result, we choose three layers of SVR model.

We compare our method with several representative learning-based SR algorithms, including the sparse representation based algorithm (ScSR) [9] and the neighborhood embedding algorithm (NE) [10]. The results of those two algorithms are obtained by running the corresponding code packages. All test images are from ASDS-AR [11] algorithm code package. The proposed algorithm uses one of the test images as training image. Here, we use the image $girl$ to train a multi-layer SVR model. To reduce training time and select different types of points, we select a point every ten points, rather than select all points in the training image.

Fig. 3 and Fig. 4 show the comparison of the proposed algorithm and several other representative algorithms. We see that the reconstructed HR images by method [9] have many jaggy and ringing artifacts. The NE algorithm [10] is effective in suppressing the ringing artifacts, but it generates pixel-wise constant block artifacts. The results of the proposed method is best in visual quality. The reconstructed edges are much sharper than all the other four competing methods, and more image fine structures are recovered.

The PSNR and SSIM values of the reconstructed HR images are shown in Table 1. Although the performance differences among these methods are small, we can still observe

Table 1. The PSNR (dB) and SSIM results (luminance components) of reconstructed HR images.

Image	Bicubic		ScSR [9]		NE [10]		SVR($N = 1$)		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bike	32.0178	0.8006	30.9291	0.7225	31.3087	0.7554	32.3433	0.8174	**33.0135**	**0.8459**
Butterfly	33.2577	0.8822	32.0631	0.8078	31.4058	0.8263	33.4627	0.8933	**34.2422**	**0.9130**
Raccoon	33.7221	0.8052	31.9906	0.7250	33.5754	0.7770	34.4082	0.8332	**34.8824**	**0.8530**
Flower	33.8980	0.8402	32.6192	0.7756	33.7389	0.8288	34.5825	0.8658	**35.1435**	**0.8856**
Girl	36.3413	0.8071	34.2435	0.7254	**37.3866**	0.8310	37.0175	0.8476	37.2677	**0.8544**
Hat	35.9473	0.8774	34.6495	0.8238	35.3740	0.8571	36.4337	0.8900	**36.7849**	**0.9014**
leaves	32.2696	0.8745	31.4654	0.8184	30.7650	0.8289	32.3956	0.8882	**33.0416**	**0.9147**
Parents	35.4741	0.8958	34.2294	0.8514	34.9153	0.8776	36.1603	0.9114	**36.7544**	**0.9243**
Parrots	36.4792	0.9032	35.2049	0.8525	36.1631	0.8865	37.2322	0.9157	**37.5803**	**0.9527**
Plants	36.9490	0.8969	35.4003	0.8502	36.8124	0.8924	37.5755	0.9201	**38.3119**	**0.9319**

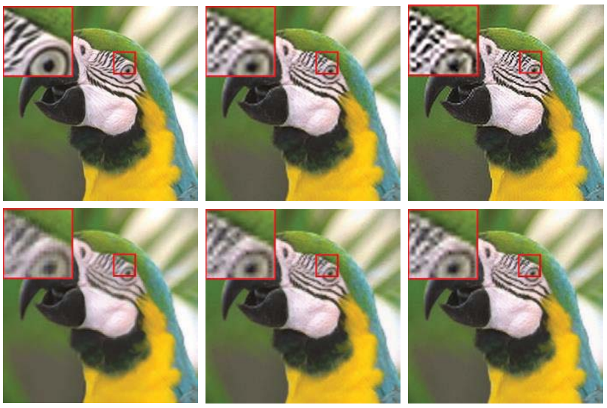


Fig. 3. Reconstructed HR images of *Parrots* by different methods. Top row: Original, Bicubic, SCSR. Bottom row: NE, SVR ($N = 1$), proposed ($N = 3$).

Fig. 4. Reconstructed HR images of *flower* by different methods. Top row: Original, Bicubic, SCSR. Bottom row: NE, SVR ($N = 1$), proposed ($N = 3$).

that the proposed method constantly outperforms the competing methods. Also, our multi-layer SVR model can be processed offline and be called during the test process, which greatly reduce the computation cost.

5. CONCLUSION

In this paper, we propose a novel image SR approach based on multi-layer SVR model to restore as much detailed information as possible by building different mapping relationships. Specifically, it constructs the SVR model not only between LR and HR images, but also between the LR image and the error image. Pixel-wise classification is introduced to divide pixels into different classes such as horizontal edges, vertical edges and smooth areas, which is more conductive to highlight the local characteristics of the image. Moreover, the input elements of SVR model are weighed respectively on the basis of the space position of corresponding pixel in the HR image patch. Comparison with existing algorithms shows the advantages of the proposed method.

6. ACKNOWLEDGEMENTS

We want to thank the helpful comments and suggestions from the anonymous reviewers. This research was supported partially by the National Natural Science Foundation of China (Nos. 61125204, 61125106, and 61101250), the Program for New Century Excellent Talents in University (No. NCET-12-0917), the Program for New Scientific and Technological Star of Shaanxi Province (No. 2012KJXX-24), the National High Technology Research and Development Program of China (No. 2013AA01A602), the Program for Changjiang Scholars and Innovative Research Team in University of China (No. IRT13088), and the Shaanxi Innovative Research Team for Key Science and Technology (No. 2012KCT-02).

7. REFERENCES

[1] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[2] Karl S Ni and Truong Q Nguyen, "Image superresolution using support vector regression," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1596–1610, 2007.

[3] Dalong Li and Steven Simske, "Example based single-frame image super-resolution by support vector regression," *Journal of Pattern Recognition Research*, vol. 5, no. 1, pp. 104–118, 2010.

[4] Chih-Chung Chang and Chih-Jen Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27, 2011.

[5] Nello Cristianini and John Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge university press, 2000.

[6] V Vapnik, *Statistical learning theory*, Wiley, New York, 1998.

[7] Emmanuel Vazquez and Eric Walter, "Multi-output support vector regression," in *Proc. of IFAC Symposium on System Identification*, 2003, pp. 1820–1825.

[8] Shuo Xu, Xin An, Xiaodong Qiao, Lijun Zhu, and Lin Li, "Multi-output least-squares support vector regression machines," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1078–1084, 2013.

[9] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[10] Hong Chang, Dit-Yan Yeung, and Yimin Xiong, "Super-resolution through neighbor embedding," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 275–282.

[11] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.