Single Image Super Reslution Using Fixed Budget- Kernel Recursive Least Square

[a] Jesna Anver, [b]Abdulla P
[a] Associate Professor, CSE, Toc H Institute of Science and Technology, Cochin, India
[b]Professor, Division of Electronics, School of Engineering, CUSAT, Cochin, India

Abstract

Image super resolution using Learning based method uses training samples from the existing low and high resolution to produce the high resolution image. This paper brings a solution using fixed-budget kernel recursive least square, which finds a dictionary based non linear relation from the low resolution and high resolution training set of the image. Fixed-budget kernel recursive least square is an online algorithm that removes the irrelevant data from the dictionary during training phase and constructs the model for HR image prediction. The result is compared to the existing super resolution algorithms and provides an improved performance.
KEYWORDS— Image super resolution, Kernel machine, Fixed budget-Kernel recursive least squares.

I. Introduction

Super resolution (SR) is a method of finding a high resolution (HR) image from the existing low resolution image (LR). The super resolution algorithms are classified into two; multi-image super resolution and single image super resolution. The multi-image SR algorithms generates HR image from multiple low resolution images of the same scene and the single image SR algorithms generates HR image from a single image input. The learning based super resolution algorithms [1-5], determines the regression between LR and HR training data. The high frequency information is recognized from the training set. These SR algorithms develops a linear regression for determining the high resolution image. The support vector regression super resolution (SVR-SR) proposed by [6] finds a nonlinear function. SVR-SR, initially considers all training samples for the SVR expression and after solving the nonlinear expression the sparse training samples are eliminated, whose coefficients vanish. Hence the complexity is the same. In [7] a nonlinear dictionary based solution had been developed using kernel recursive least squares but the subset do not eliminate the irrelevant data. Here we use fixed budget kernel recursive last square (FB-KRLS) proposed by[8] , which takes only the relevant subset of training samples and constructs the non-linear regression.

II. Fixed Kernel Recursive Least Squares

The machine learning algorithms builds regressions on a transformed data from the input space to the Hilbert space (feature space F), i.e x is mapped to $\phi(x_i)$ in F. The kernel functions computes the inner products of feature vectors without the knowledge of the feature vectors themselves, by evaluating the kernel function:

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

where, x_i, x_j denote the input vectors and ϕ is the mapping onto the F space. The commonly used kernel functions are the Gaussian kernel with variance σ^2 is expressd as, $k(x_i, x_j) = \exp\{-\|x_i - x_j\|^2 / 2\sigma^2\}$ and the polynomial kernel of degree p is,

$k(x_i, x_j) = (a\langle x_i, x_j\rangle + b)^p$. The Representer theorem states the solution attained by the kernel methods is represented as $f(\mathrm{x}) = \sum_{i=1}^{t} \alpha_i k(\mathrm{x}_i, \mathrm{x})$, where f is a linear predictor in Hilbert space.

Kernel-based least-squares (LS) problem operates on a training set $Z_t = \{x_i, y_i\}_{i=1}^{t}$, where (x_i, y_i) are the input, output pairs. Here a non-linear predictors $\hat{y}_t$ of y_t, given $Z_{t-1} \cup \{x_t\}$ is obtained by minimizing the mean squared error over the training data. $\min_{\alpha} |y - K\alpha|^2 + \lambda\alpha^T K\alpha$ .where $y \in {}^{M\times 1}$ is the output vector, $K \in \mathbb{R}^{M\times M}$ the kernel matrix and λ is the regularization factor. The solution is $\alpha = (K + \lambda I)^{-1} y$.

The FB Kernel RLS prunes least significant data from the memory. Hence FB Kernel RLS belongs to *active learning* algorithms. The absolute coefficient $|\alpha_i|$ of the pruned data x_i will have least value and the pruned data has least error to the regression.

III.The Algorithm

The low resolution patches I_{LR} of size L×L is preprocessed using the expression $x = vectorize(I_{LR}) - centerpixel(I_{LR}) \in \mathbb{R}^{L^2}$ .The image is super resolved by a factor U . The low resolution image I_{LR} is classified in to 'n' classes using clustering algorithm. The estimator for each class, implies each pixel in the class generates four output pixels. The formulation for each class of pixel then becomes,

$$\min_{\alpha_n} |y_n - K_n\alpha_n|^2 + \lambda\alpha_n^T K_n \alpha_n^T \qquad (1)$$

the solution is $\alpha_n = (K_n + \lambda I)^{-1} y_n$ (2)

of the current designations.

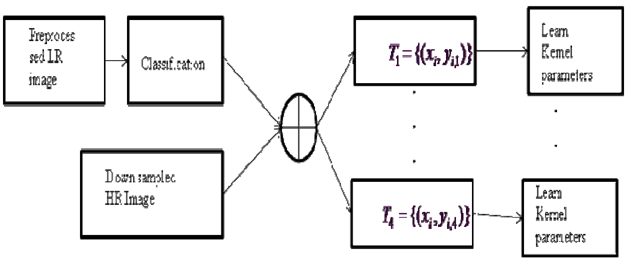

Figure 1: Training phase

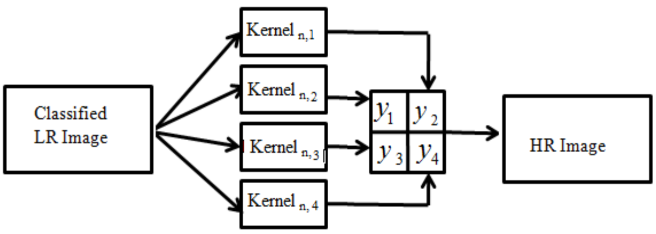

Figure 2: Testing phase

The figure 1 shows the training phase of the super resolution algorithm. The input low resolution is first preprocessed and is then classified in to n classes. The training pairs include the pixels from down sampled high resolution image and classified input image, at same pixel position. The training pairs are used to learn the FB Kernel RLS parameters. The learned Kernel parameters are used in the testing phase as shown in figure 2 to generate the output pixels for the high resolution image.

IV. Results and Discussions

The preprocessing is performed on a patch size of 3x3. The center pixels are normalized and the super resolution is applied for a factor of U=2. The images taken for analysis are the standard images in the internet. The algorithm is developed in Matlab and tool for FB Kernel RLS is taken from [8]. The taken Gaussian kernel is applied here as it groups the kernel that is closer. The optimal solution is taken by performing five fold cross validation. The results are compared with kernel based support vector machine (SVR) based image super resolution proposed by [6] and the KRLS solution [7]. The table 1 shows PSNR values and time required to super resolve the images from 64 x 64 size to 128 x 128 size. The proposed algorithm gives a better PSNR and less time when compared to SVR method.

Methods	Butterfly Image	Mandril Image	Bear mage
SVR-SR	27.963	28.54	28.1
KRLS-SR	28.57	29.2	28.84
FB-KRLS-SR	28.71	29.66	29.06

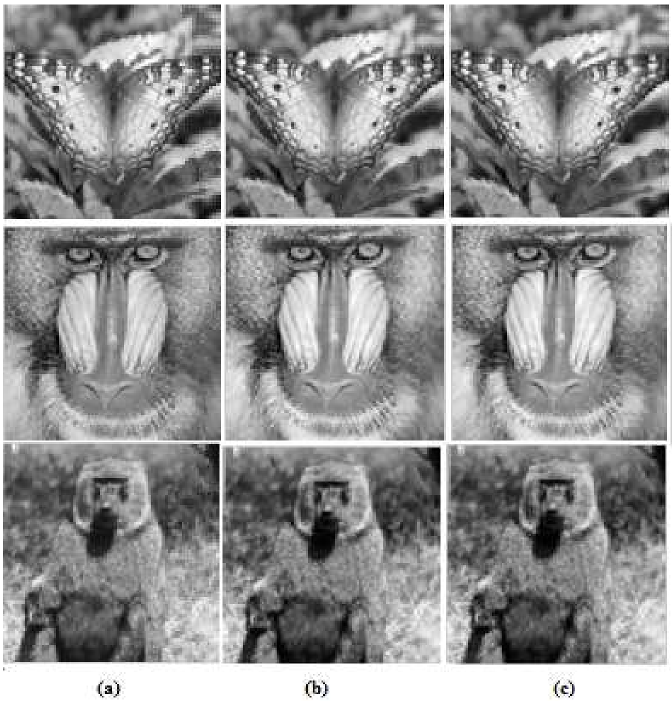

Fig. 1. (a) SVR-SR (b) KRLS-SR (c) FB-KRLS-SR

V. Conclusion

A nonlinear based learning algorithm is proposed using FB Kernel RLS. Rather than eliminating insignificant training samples after calculation of kernel parameters, FB Kernel RLS super resolution eliminates the least significant training samples before evaluation of the non linear expression and hence the time required for super resolution is minimized.

References

[1] H, Chang., D, Y, Yeung and Y, Xiong., “Super-resolution through neighbor embedding”. In CVPR, volume 1, pages 275–282, 2004.

[2] J, Sun., N. N. Zheng., H, Tao. and H, Y, Shum., “Image hallucination with primal sketch priors”. In CVPR, volume 2, pages 729–736, 2003.

[3] Kong, D., Han, M., Xu, W., Tao, H., Gong, Y, “A conditional random field model for video super-resolution”. In: Proceedings of IEEE International Conference on Pattern Recognition, China.

[4] W, T, Freeman., E, Pasztor. and O, Carmichael., “Learning low-level vision”. International Journal of Computer Vision", pages 25–47, 2000

[5] R, Fattal, “Image upsampling via imposed edge statistics. ACMTransactions on Graphics”, 26(3):95:1–95:8, 2007

[6] K, S, Ni. and Truong, Q, Nguyen, “Image Superresolution Using Support Vector Regression”. IEEE Transaction on Image Processing, vol 16 no.6 pp 1596-1610, 2007.

[7] Jesna Anver, Abdulla P, “Single-image super-resolution using kernel recursive least squares”, Springer- signal image and video processing, vol. 10, no. 8, pp. 1551–1558, Nov. 2016.

[8] S. Van Vaerenbergh, I.Santamaria. W. Liu, and J.C Principe, “Fixed-budget kernl rcursive least-squares”,2010, IEE International Conference on Acoustics, Spech and signal processing (ICASSP), 2010

[9] Steven, Van, Vaerenbergh., 2010. Kernel Methods Toolbox (KM-BOX): a MATLAB toolbox for nonlinear signal processing and machine learnin. Software available athttp://sourceforge.net/p/kmbox.

.