Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011

Improved Image Super-Resolution by Support Vector Regression

Le An and Bir Bhanu

***Abstract*—Support Vector Machine (SVM) can construct a hyperplane in a high or infinite dimensional space which can be used for classification. Its regression version, Support Vector Regression (SVR) has been used in various image processing tasks. In this paper, we develop an image super-resolution algorithm based on SVR. Experiments demonstrated that our proposed method with limited training samples outperforms some of the *state-of-the-art* approaches and during the super-resolution process the model learned by SVR is robust to reconstruct edges and fine details in various testing images.**

I. Introduction

With the wide spread application of video cameras, surveillance systems and hand-held devices that are equipped with moderate image sensors, it is desirable to generate images or video streams with high quality while not increasing the cost of the hardware. The imaging process of these sensors can be modeled by

$$I_{LR} = DHI_{HR} + n \tag{1}$$

where I_{HR} is the high-resolution image that undergoes blurring (H) and downscaling (D) procedures with additive noise n. The output is a low-resolution image I_{LR} that we often observe. Normally the noise n is assumed to be white Gaussian noise. Fig. 1 illustrates this process. Often during this process, the details of the image (high-frequency component) are lost.

The purpose of image super-resolution (SR) is to reverse this process to recover the high-resolution (HR) image from the low-resolution (LR) observations. SR has other names such as image scaling, image interpolation and enlargement. Note that SR is generally an ill-posed problem due to insufficient prior information about the contents of the HR images (i.e. by downscaling a HR image, the possible LR image is not unique).

Inspired by the pioneer work of Tsai and Huang [1], there has been extensive work in image and video SR. There are different approaches for image SR. The first type of SR algorithms requires multiple LR images from the same scene (i.e., consecutive frames taken from a video stream) as input, then all of those images are registered and fused to generate super-resolved images based on the assumption that each LR image contains relevant yet slightly different information that can contribute to the HR reconstruction.

Another type of SR algorithms is single image based interpolation. The well-known techniques such as bicubic interpolation [2] are easy to implement and fast in processing. However, interpolation often gives over smooth results due to its incapability to reconstruct the high-frequency components of the desired HR image. This could be solved by exploiting the natural image priors such as local structure gradient profile priors [3]. The disadvantage for this kind of approaches is that the heuristics about natural images are made, which would not always be valid and for images with fine textures the reconstructed HR image may have the water-color like artifacts.

Fig. 1. Imaging process from high-resolution image to low-resolution observation. (a) The original image. (b) Blurred image. (c) Downscaled low-resolution image. (d) Low-resolution image corrupted by noise.

Recently learning based approaches have been proposed for image SR [4] [5] [6]. In [4] an example based learning algorithm is proposed by predicting the HR images from LR images via a Markov Random Field (MRF) model that is computed by belief propagation. In [5] support vector regression (SVR) is applied to single image super-resolution in Discrete Cosine Transform (DCT) domain. In [6], the SVR is applied to find the mapping between the LR images and the HR images in the spatial domain. In our approach, the SR is also formulated as a regression problem which is solved by SVR in the spatial domain. However, there are

L. An is with the Center for Research in Intelligent Systems, Electrical Engineering Department, University of California at Riverside, Riverside CA 92521 USA (e-mail: lan004@ucr.edu).

B. Bhanu is with the Center for Research in Intelligent Systems, Electrical Engineering Department, University of California at Riverside, Riverside CA 92521 USA (e-mail: bhanu@cris.ucr.edu).

distinctions between our approach and the approach in [6]. *First*, in our approach we do not aim at estimating the high-frequency component of the LR image to be super-resolved. Instead, the prediction of our algorithm is the pixel value itself. *Second*, the feature vectors that we choose not only contain the pixel values from a neighborhood but also the local gradient information. *Third*, the neighboring pixel values are assigned with different weights because they do not contribute equally to generate the output pixel in the super-resolved image. Furthermore, in the training process we use images of small sizes only to form a relatively small training dataset for efficiency consideration. The experiments show that even with a small training set our method can still generate visually pleasing results.

The rest of the paper is organized as follows: The support vector regression is briefly reviewed in section II. Section III introduces the proposed algorithm for image super-resolution. Section IV shows the experimental results and gives analysis and comparison to other SR algorithms. Finally, Section V concludes the paper.

II. SUPPORT VECTOR REGRESSION

Suppose we have a training set $\{(x_p, y_p)\}_{p=1}^{n}$ where x_p is the feature vector and y_p is the corresponding observation. Traditional linear regression which seeks a linear function $f(x) = <w, x> + b$ ($<\cdot,\cdot>$ denotes the dot product) that minimizes the mean square error is often not capable of separating the nonlinearly distributed input data while on the other hand by using a transformation function $\phi(x)$, the data is mapped into a higher dimensional feature space in which the data becomes separable. The nonlinear SVR solves the following optimization problem:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{p=1}^{n}(\xi_p + \xi_p^*) \tag{2}$$

$$\text{subject to} \begin{cases} y_i - (<w, \phi(x_p)> + b) \le \varepsilon + \xi_p, \\ (<w, \phi(x_p)> + b) - y_i \le \varepsilon + \xi_p^*, \\ \xi_p, \xi_p^* \ge 0, i = 1, \ldots n \end{cases}$$

where ξ and ξ^* are the slack variables, C is a constant and determines the trade-off between the flatness of the mapping function and the amount up to which deviations larger than ε are tolerated. The model generated by SVR depends only on a subset of the training data since the cost function ignores the training data within the threshold of ε. The function $k(x_p, x_q) = <\phi(x_p), \phi(x_q)>$ is called the kernel function. A nice tutorial on SVR can be found in [7].

In our approach, x_p comes from the initial estimation of the LR image and y_p is from the corresponding HR image. Then a model is learned by SVR. In the prediction process, the learned model will be applied to the input LR image to generate a super-resolved HR image.

III. SUPER RESOLUTION ALGORITHM

The algorithm consists of two processes: training and prediction, as shown in Fig. 2.

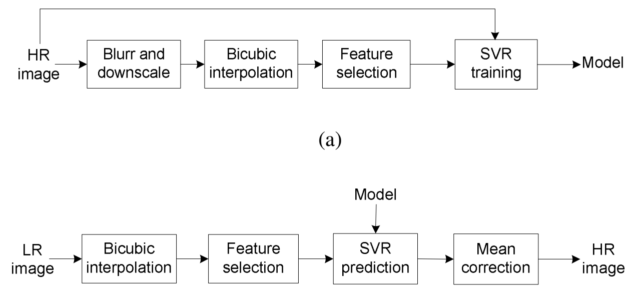

Fig. 2. (a) Training process. (b) Prediction process.

In the training process, we first blur the HR images and then downscale them by a factor of 2 to create the LR images. An initial estimation of the HR image is carried out using bicubic interpolation with an upscaling factor of 2 on the generated LR image. For each pixel at location (i, j) in the upscaled image, we take a local image patch of size $m \times m$ centered at (i, j). This image patch is then weighted by a matrix of the same size. This matrix is constructed from a 2-D Gaussian distribution that assigns largest weight to the pixel at (i, j) and smaller weight to the other pixels that are further away from the center pixel in the local patch. The weighted image patch is then converted to a row vector:

$$x_{i,j} = vec(W_G(R_{i,j} I_{BI})) \tag{3}$$

where I_{BI} is the bicubic interpolated image and $W_G(R_{i,j} I_{BI})$ is the weighted local image patch taken at (i, j) by the patch extraction operator $R_{i,j}$. Function vec reshapes the matrix into a row vector $x_{i,j}$ of length m^2.

The gradient of the bicubic interpolated image is calculated in both horizontal and vertical direction at each pixel.

$$\begin{aligned} g_h(i, j) &= \frac{1}{2}((I_{i,j+1} - I_{i,j}) + (I_{i+1,j+1} - I_{i+1,j})) \\ g_v(i, j) &= \frac{1}{2}((I_{i+1,j} - I_{i,j}) + (I_{i+1,j+1} - I_{i,j+1})) \end{aligned} \tag{4}$$

$I_{i,j}$ is the pixel value at (i, j). The horizontal gradient magnitude $g_h(i, j)$ and the vertical gradient magnitude $g_v(i, j)$ are concatenated to the row vector $x_{i,j}$.

For each pixel in the initially interpolated image at (i, j), $x_{i,j}$ is now a $m^2 + 2$ dimensional feature vector. The corresponding observation $y_{i,j}$ is the pixel value at position (i, j) in the HR image. We supply SVR with all the feature vectors constructed from the training dataset and their corresponding observations. The generated model is then saved for the future use.

In the prediction process, we first upscale the testing LR image also using bicubic interpolation by the same factor of 2. For each pixel in the interpolated image, the local image patch of the same size is taken and the image gradient in two directions is calculated to get the features vectors in the same manner as we do in the training process. Now the output image z is constructed. The last step is to correct the mean of z since the mean of the upscaled image should be preserved to be the same as that of the input LR image due to the unchanged image structures and contents in the upscaling process. The pixel value of the final output at (i, j) is:

$$\hat{z}_{i,j} = \frac{m_{LR}}{m_z} \cdot z_{i,j} \tag{5}$$

where m_{LR} is the mean pixel value of the LR image and m_z is the mean pixel value of z.

IV. Experiments

A. Experimental setup

For the implementation of SVR we use LibSVM [8]. We choose Gaussian function as the kernel function. The parameters in the SVR are selected by cross-validation ($C = 362$, $\varepsilon = 2$ and the standard deviation $\sigma = 1$ in the Gaussian kernel function). The downscaling factor for the HR images and the corresponding upscaling factor for the LR images are both 2. For both training and testing we only consider the luminance component of the images.

Fig. 3. Images for training. From left to right and top to bottom: Lena, Peppers, Man, Lake, Truck and Car.

We use 6 images for training and 9 images for testing. All the images used in the training and testing processes are originally taken from the USC-SIPI Image Database [9]. The HR images we used in the training set are all of size 128×128. The LR images are blurred by a 3×3 uniform point spread function (PSF) and then downscaled in order to get the LR images for training. The LR images we used in training are of size 64×64 while for the testing purpose the input LR images are of size 128×128. The HR images of size 256×256 as the ground truth are used to later evaluate the performance quantitatively. The size of the image patch we use to get feature vectors is 5×5. We do not add noise deliberately to the LR images because in reality for the case of image SR, the input LR image is not necessarily corrupted by noise and even if it is the case, a pre-processing with a robust image denoising algorithm can effectively remove the noise [10].

Both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [11] are used to measure the quality of the super-resolved images compared to original HR images. PSNR between two images of size $M \times N$ is calculated by

$$PSNR = 10 \log_{10} \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} 255^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (x(i,j) - y(i,j))^2} \tag{6}$$

where 255 is the maximum possible gray pixel intensity value, $x(i, j)$ and $y(i, j)$ are the pixel values at the same location (i, j) from image x and y.

SSIM is designed to better match the human perception compared to PSNR, which sometimes is inconsistent with the visual observation [11]. SSIM is defined as:

$$SSIM = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{7}$$

where x and y are the two images to be compared, μ and σ are respectively the average and variance of the pixel values of the images x or y. σ_{xy} is the covariance of x and y. The value of SSIM is a scalar less than or equal to 1 and 1 means the two images in comparison are exactly the same.

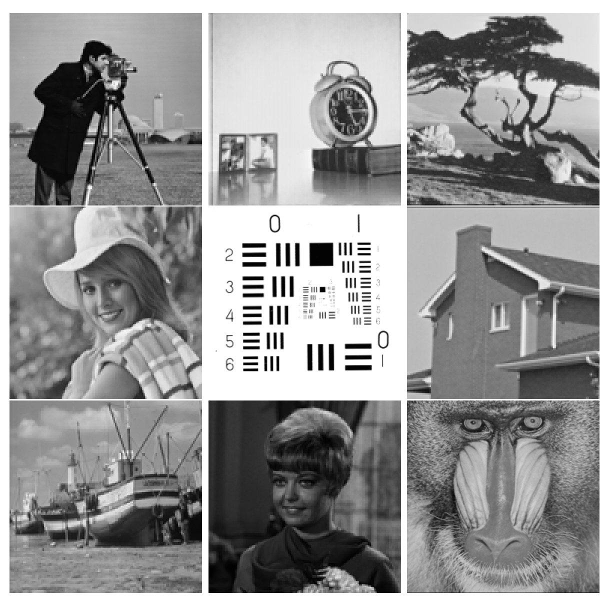

Fig. 4. Images for testing. From left to right and top to bottom: Cameraman, Clock, Tree, Elaine, Synthetic, House, Boat, Girl and Mandrill.

B. Results

First we validate our choice of feature selection (pixel intensity values and gradient magnitude) by comparing the result of the proposed method to the output of SVR with feature vectors that only contain the pixel values as similarly proposed in [6]. Fig. 5 gives an illustrative comparison from a test image.

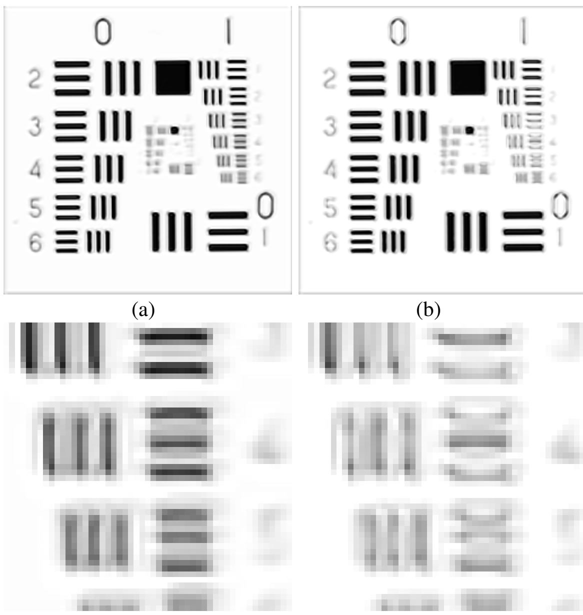

Fig. 5. (a) Result by proposed method. (b) Result by SVR without gradient information. (c) Magnified part from (a). (d) Magnified part from (b).

It can be seen that by adding image gradient information to the feature vectors, the model learnt by SVR is capable of reconstructing edges and fine details from LR images.

We compare the proposed super-resolution algorithm with *state-of-the-art* methods including sparse coding [12] and kernel regression (KR) [13] with the implementations provided by the authors of [12] and [13]. Also results by bicubic interpolation (BI) [2] are provided as reference. We do not compare our results to [6] directly since without knowing the parameter settings in their SVR training we are not able to produce results with their algorithm exactly. However as indicated by Fig. 5 the proposed algorithm is better at handling details and edges.

Table I and II show the PSNR and SSIM results for the test images. As shown, our method outperforms the other methods with respect to both PSNR and SSIM. Note that the reference methods are specifically designed for single image super-resolution. We also show some results in Fig. 6. **It is suggested to view the results on your computer.**

By visual inspection, our method produces sharp images. While sparse coding [12] is good at preserving high-frequency details, the generated images suffer from noticeable artifacts especially along the edges and image boundaries. Bicubic interpolation [2] and kernel regression [13] produce outputs without many high-frequency components. Kernel regression generates ghost artifacts, for example on the upper tripod in "Cameraman". The model learned by SVR in our method is able to generate fine details and sharp edges which lead to better visual quality. Both objective evaluation (by PSNR and SSIM) and subjective evaluation confirm the advantages of our method.

TABLE I
PSNR COMPARISON

Image	BI[2]	SC[12]	KR[13]	Proposed
Cameraman	33.07	32.80	32.71	34.54
Clock	33.71	33.34	33.25	35.53
Tree	31.83	31.59	31.20	33.33
Elaine	35.24	34.55	33.72	37.59
Synthetic	31.57	32.05	29.83	32.15
House	34.57	33.80	33.71	36.26
Boat	32.49	32.33	31.72	33.95
Girl	36.26	36.21	35.34	38.37
Mandrill	30.89	31.00	30.35	31.62

TABLE II
SSIM COMPARISON

Image	BI[2]	SC[12]	KR[13]	Proposed
Cameraman	0.818	0.835	0.756	0.864
Clock	0.904	0.887	0.869	0.927
Tree	0.811	0.819	0.736	0.864
Elaine	0.921	0.916	0.860	0.940
Synthetic	0.855	0.882	0.647	0.896
House	0.863	0.845	0.819	0.885
Boat	0.799	0.838	0.707	0.856
Girl	0.904	0.912	0.857	0.923
Mandrill	0.648	0.738	0.551	0.744

V. CONCLUSIONS

In this paper, an algorithm for single image super-resolution based on support vector regression is proposed. By combining the pixel intensity values with local gradient information, the learned model by SVR from low-resolution image to high-resolution image is useful and robust for image super-resolution. We conducted the experiments on different types of images and the results are promising. Furthermore, the size of the training set is limited which makes the training relatively fast while still achieving good results. By comparing our method to the previous works, we find out that our method is able to produce better super-resolved images than *state-of-the-art* approaches. We believe that by selecting more informative features besides pixel intensity and gradient, the result can be further improved. By adopting a larger and more comprehensive image dataset for training, the generated model would yield better results for image super resolution.

REFERENCES

[1] R. Tsai, T. Huang, "Multi-frame image restoration and registration," *Advances in Computer Vision and Image Processing*, vol. 1, no. 2, JAI Press Inc., Greenwich, CT, 1984, pp. 317–339.

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

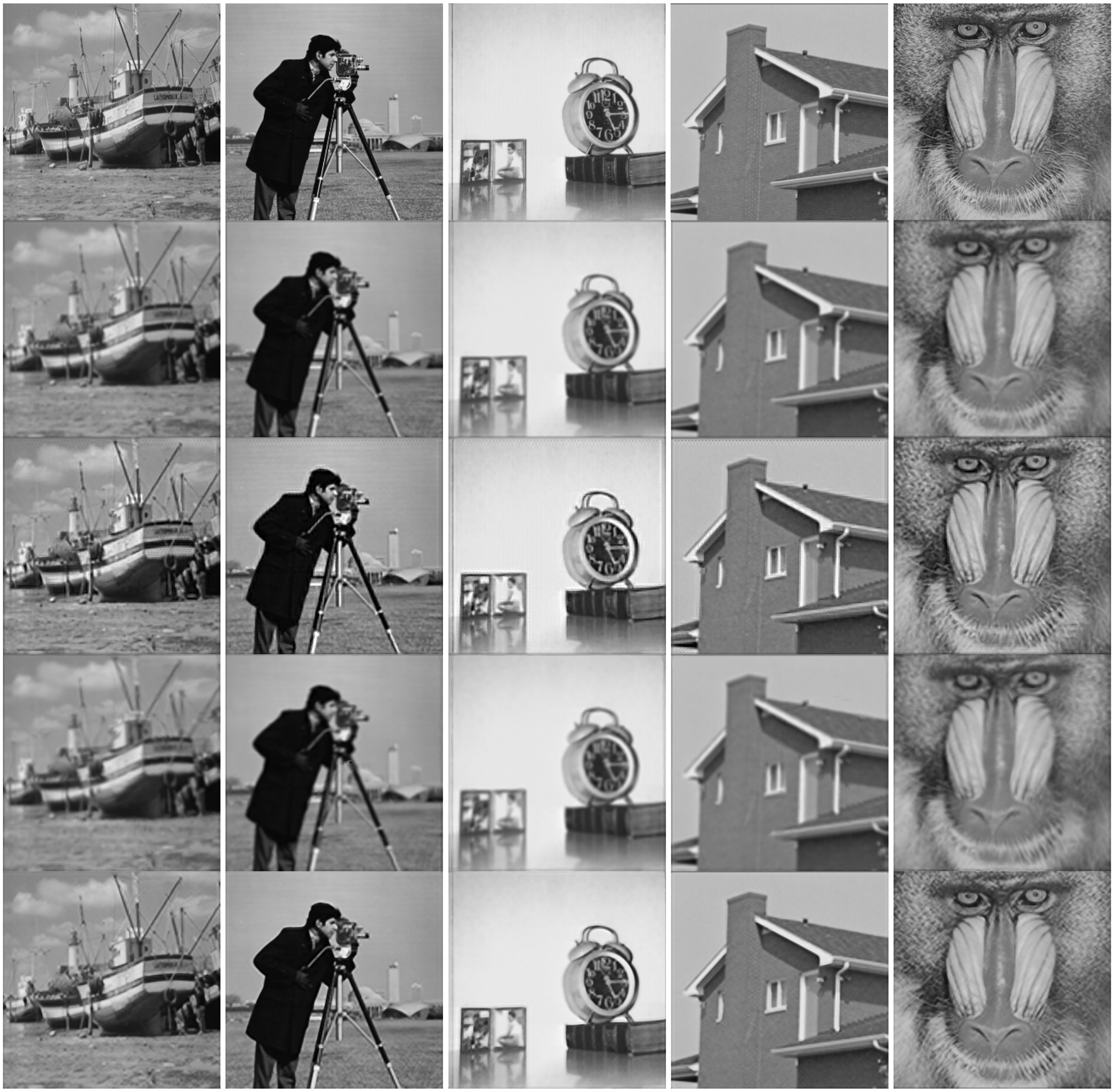

Fig. 6. From top to bottom: Original HR images, results by bicubic interpolation [2], results by SC [12], results by KR [13], results by the proposed method.

[3] J. Sun, Z. Xu, and H.-Y. Shum, “Image super-resolution using gradient profile prior,” *In Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[4] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning low-level vision,” *International Journal of Computer Vision*, 40 (1), pp. 25-47, 2000.

[5] K. Ni, S. Kumar, N. Vasconcelos, and T. Q. Nguyen, "Single image superresolution based on support vector regression", *In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 601-604, 2006.

[6] D. Li, S. Simske, and R. Mersereau, “Single image super-resolution based on support vector regression”, *In Proc. of International Joint Conferences on Neural Networks*, pp. 2898-2901, 2007.

[7] A. J. Smola, and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing 14*, pp. 199–222, 2004.

[8] C. -C. Chang, and C. -J. Lin, “LIBSVM: a library for support vector machines,” 2001. *Software available at:* http://www.csie.ntu.edu.tw/~cjlin/libsvm

[9] *Available at*: http://sipi.usc.edu/database/

[10] A. Baudes, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” *In Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 60-65, vol. 2, 2005.

[11] 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.

[12] J. Yang, J. Wright, T. Huang, and Y. Ma. “Image Super-resolution as Sparse Representation of Raw Image Patches,” *In Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[13] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 349-366, 2007.