Journal of Pattern Recognition Research 1 (2010) 104-118

Received September 30, 2010. Accepted October 9, 2010.

Example Based Single-frame Image Super-resolution by Support Vector Regression

Dalong Li *dalong.li@hp.com*
Hewlett-Packard
11445 Compaq Center Dr. West. Houston, TX 77070, USA

Steven Simske *steven.simske@hp.com*
Hewlett-Packard
3404 East Harmony Road, Fort Collins , CO 80525, USA

Abstract

As many other inverse problems, single-frame image super-resolution is an ill-posed problem. The problem has been approached in the context of machine learning. However, the proposed method in this paper is different from other learning based methods regarding how the input/output are formulated as well as how the learning is done. The assumption behind example based methods is the local similarity across seemingly different images. The assumption is illustrated by examples of image coding. Because of the differences in formulating the input/output and the implementation of Support Vector Regression (SVR), it is shown that the proposed approach outperforms the competing SVR method and the kernel regression method in terms of Peak Signal-to-Noise Ratio (PSNR), objective measurements of image quality. Since example based approaches are based on training, in which we know exactly what the output shall be. Therefore, it is proper to objectively measure the performance since the trained model is expected to "correctly" restore the image rather than to enhance the image, e.g. sharpening.

Keywords: Support Vector Regression, single-frame image super-resolution, ill-posed problem, example-based, machine learning.

WWW.JPRR.ORG

1. Introduction

Super-resolution techniques estimate an image at higher resolution from its low-resolution observations. It has found useful in many applications, such as video surveillance and automatic target recognition [1, 2]. The super-resolution problem often implies multiframe super-resolution, where a high-resolution image is obtained by combining the non-redundant information found in multiple low-resolution frames. Naturally, image registration and image reconstruction are the common two major steps in multiframe super-resolution algorithms. In this paper, we are concerned with single-frame super-resolution (upscaling), which is different from the multiframe case. There is only one observed low resolution image. The problem is highly ill-posed. To explain why it is ill-posed, we can view that a low resolution image is obtained by smoothing on the higher resolution image followed by a downsampling. The smoothing step is to prevent image aliasing. The goal of single frame superresolution is to estimate the high resolution image based on the observed low resolution image. Superresolution is to reverse the anti-aliasing and downsampling process. It is an inverse problem. Considering the anti-aliasing, an observed pixel in low resolution image can be viewed as a weighted sum of pixels from high resolution image pixels. The number of unknowns is more than the number of constraints. In this regard, like many other inverse problems such as blind image deconvolution, super-resolution is an ill-posed problem. There is no solution unless additional constraints are introduced to the

© 2010 JPRR. All rights reserved. Permissions to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to republish, requires a fee and/or special permission from JPRR.

problem. A commonly used constrain is the smoothness of an image. In multiple frames super-resolution, more constraints are given, but the nature of problem is still ill-posed since there are different models which lead to different solutions.

Single-frame super-resolution has a number of applications such as magnification (zooming, enlargement, etc.) of images sampled below the Nyquist rate. From a low-resolution image, interpolation algorithms can generally fill the missing pixel values on a finer grid. Bilinear, bicubic, or B-spline kernels [3] are among those commonly used interpolation techniques. Those interpolation algorithms assume the local smoothness of the image. Consequentially, the interpolated image is blurry. Recently, kernel regression [4] was proposed as an effective tool for image denoising, upscaling, interpolation, fusion, and more. The assumption in this method is still smoothness of image. This kernel regression based method is compared in this work.

Because of its ill-posed nature, additional information is needed such as a training set where the ground-truth high resolution image is available [5, 6, 7, 8, 9]. We had reported our previous work in applying support vector regression to a number of image processing tasks: blind image deconvolution [10, 11], image denoising [12]. Recently, Ni et al. [13] also reported their work in using SVR for image super-resolution. Learning based methods differ in two aspects: how the input and output are formulated and how the learning is done. Freeman et al. [6] formulated a low-resolution patch together with its neighboring high-resolution patch as input and the corresponding high-resolution patch as output. Regarding the learning, the authors explored two different approaches to exploit neighborhood relationships in single-frame super-resolution algorithms. The first one uses a Markov network to probabilistically model the relationships between high- and low-resolution patches, and between neighboring high-resolution patches. The first approach uses an iterative algorithm, which converges very quickly. The second approach is a one-pass algorithm that uses the same local relationship information as the Markov network. It is a fast, approximate solution to the Markov network method. Because it is a patch to patch mapping, the boundary artifact is commonly observed, i.e., the predicted high-frequency patch is not compatible with those of the neighboring patches. To overcome the boundary effects, both the low-frequency patch and the overlapping high-frequency are used in searching the best match in the training set.

In [13], the input is a low-resolution patch with the central pixel in it removed and the output is $U \times U$ pixels in high-resolution image. U is the factor by which the image size is increased, for example, U is 2 when the size of an image is to be doubled in each dimension. There are several issues in such input and output arrangements. First of all, the central pixel in the low-resolution patch is the most important one in predicting the output. Correlation of pixels decreases as they are separated further, so the pixels on the boundary of a patch have less importance in predicting the central pixel in the corresponding high resolution patch. Secondly, the outputs are 4 (assuming a factor of 2 by 2) pixel values. Therefore, 4 SVRs are needed with exactly the same input vector as against to 1 SVR in our case. The differences lead to performance difference. As shown in Section 3, our SVR achieved higher ISNR (improvement in signal-to-noise ratio) with much less training.

The input to our SVR based image super-resolution is the interpolated low-resolution patches and the output is the set of the central pixels. Instead of a patch-to-patch mapping, it is patch-to-pixel. The boundary artifacts in patch-to-patch mapping method is avoided. The input patches are largely overlapped, therefore the processed image is guaranteed to be smooth.

In this paper, the assumptions of the example-based methods in general is clarified. SVR is one of those example-based approaches. No explicit modeling is utilized. SVR predicts output based on support vectors (SVs) that are identified from all the training samples during the training phase. Though training images are different from test images, they are similar locally. That is, locally a small patch from one image can be approximated by another patch in a different image. This local similarity validates the example-based methods. To illustrate the local similarity, we demonstrated a series of image coding experiments.

The paper proceeds as follows: in Section 2, we discuss the example-based approach. The assumption is validated in 2.1. Since a SVR is chosen as the learning algorithm, a brief review of SVR is made and its architecture is illustrated in section 2.2. The architecture graphically highlights the difference between SVR and other filtering methods in image processing. Experiments are in section 3, where we compare our results with the other SVR approach introduced by Ni etc. [13] as well as the kernel regression method [4]. A conclusion is given in Section 4.

2. Super-resolution based on local patches

Example-based single frame super-resolution is a supervised learning method where the input/output are known. The examples in the training set are expected to be "similar" to those test images. Such similarity is *local* rather than *global*. The overall appearance of two images may be dramatically different, but they are locally similar. This local similarity validates the example-based methods. Given a pair of $h \times w$ training image $\{g, f\}$ where g is the observed degraded image (an interpolated image from a low resolution image in the case of single frame super-resolution) and f is the ground-truth image, a data set Ω is first created by all the patches, i.e., neighborhood of its pixels at (i, j), the $2n + 1$ by $2m + 1$ neighborhood of the blurred image $g(i, j)$ is:

$$\begin{pmatrix} g(i-n, j-m) & \dots & g(i-n, j+m) \\ \dots & g(i,j) & \dots \\ g(i+n, j-m) & \dots & g(i+n, j+m) \end{pmatrix}.$$

The matrix is vectorized either by row or by column. For example, $v(i, j) = [(g(i-n, j-m), \dots, g(i-n, j+m), \dots, g(i, j), \dots, g(i+n, j-m), \dots, g(i+n, j+m)]$ (row by row). $\Omega = \{v(i, j) | 0 \le i < h, 0 \le j < w\}$. In practice, more than one image can be used to create the ensemble Ω. Depending on the sophistication/accurancy of the learning algorithms, a certain number of images are required. Since SVR generalizes well to unseen data, the number of training images can be surprisingly low while reasonable performance is achieved. $\mathbf{G}$ is the input set in training and the output/target set $\mathbf{F}$ is made up of the corresponding central pixels at (i,j) in the ground-truth image, $\mathbf{F} = \{f(i, j) | 0 \le i < h, 0 \le j < w\}$. The mapping is patch-to-patch in [6], where $\mathbf{F}$ is made by patches. A data set $\mathbf{X}$ is made up of patches from a test image x. The goal is then to transform $\mathbf{X}$ to $\mathbf{Y}$ in the same way that maps $\mathbf{G}$ to $\mathbf{F}$. Assuming the elements in $\mathbf{X}$ and $\mathbf{G}$ are similar, this is a well-posed problem and the solution is ready. Note here the similarity is between $\mathbf{X}$ and $\mathbf{G}$, the datasets made from x and g, respectively. We are not referring to a global similarity between x and g. The following two steps are taken in an example-based approach:

- Learning the hidden mapping $f : \mathbf{G} \longrightarrow \mathbf{F}$ either explicitly or implicitly.
- Apply f in converting $\mathbf{X}$ to $\mathbf{Y}$, the desired output.

Example-based learning uses the examples for predicting the output. The learning could be as simple as nearest neighborhood rules, perhaps the simplest useful learning method. Freeman etc. [6] showed that nearest neighborhood searching is indeed a good approximation to the probabilistic Markov network in single frame super-resolution. The learning can also be rather complicated as in Support Vector Regression (SVR).

Nearest neighborhood methods view the task as a classification problem. All the examples in **G** are stored. Generally, the size of **G** is large and in order to make searching more efficient, some data structures, such as a tree, are employed [6].

SVR views super resolution as a regression problem. After SVR training, **S**, a subset of **G**, is obtained. **S** is made up of the identified support vectors. All the non-support vectors are discarded since they have no contribution in predicating the output for a given patten in test phase. The number of SVs is often much smaller than the number of the samples in the training set. Therefore, computationally SVR is more efficient in deployment in that regard. For example, the total number of samples in the Lena training set is 62,500 while the number of support vectors is only 4,310, a factor of 15 smaller.

2.1 Image Coding by Vector Quantization

Vector Quantization (VQ) based image coding is an evidence of the local similarity across images. Such local similarity is weak when the images are from very different classes, e.g. document v.s. photographic.

The coding/transform is on a patch by patch basis. To represent image A (target) by image B (source), for each patch in A, we find the best match in B and we then replace it with the matched patch. In this way, B is transformed to A.

For an 8-bit image, there are only 256 possible pixel values ([0,255]). The total number of combinations for a 4 by 4 patch is 256^4. As the size of the patch becomes smaller, the total number of combinations exponentially decreases. The smallest patch is 1 by 1. When the size of the patch is smaller, one can find a better match and the coded image has a higher Peak Signal-to-Noise Ratio (PSNR). This is confirmed by the following experiments. However, a 1 by 1 patch is not useful here, since pixel modification techniques cannot super-resolve an image.

In the experiments, there are four images: Cameraman and Lena are taken from the USC-SIPI image database [14]. Two document images are created by scanning. As the size of the patch increases, the coding error also increases. Fig. 1 shows some coding results. The in-class coding is more successful than the between-class coding. Table 1 reports the quantitative evaluations in terms of PSNR. The experiment not only demonstrates the local image similarities, but also explains why a larger size patch is often improper. A larger size of patch leads to higher coding error. In other words, the examples are less useful.

Table 1: PSNR of the coded image.

Source Image	Target Image	4 by 4	8 by 8	16 by 16
Lena	Cameraman	24.06	20.53	18.23
Cameraman	Lena	26.01	21.90	18.27
Doc1	Lena	13.32	10.25	8.23
Doc1	Doc2	27.97	21.28	17.54
Doc2	Doc1	27.72	21.06	17.26

Fig. 1: (a) The Cameraman image coded by Lena image. (b) The Lena image coded by the Cameraman image. (c) The Lena image coded by the Doc1 image. (d) The Doc1 image coded by the Doc2 image. In all of the above, patch size is 4 by 4.

2.2 Support Vector Regression

The training data is made up of input/output pairs $(X_1; y_1), \ldots, (X_l; y_l)$, where X_i is input attribute vector from an input image (an interpolated low-resolution image) and y_i are the associated output values in the ground-truth image (the high-resolution). Traditional linear regression estimates a linear function $W^T X + b$ that minimizes the mean square error:

$$\min_{w,b} \sum_{i=1}^{l} (y_i - (W^T X_i + b))^2. \tag{1}$$

To address nonlinearly distributed input data, a mapping function $\phi(x)$ is introduced in the SVM to map the data into a higher dimensional space in which the data can be linearly separated. In the high-dimensional space, overfitting occurs. To limit overfitting, a soft margin and a regularization term are incorporated into the objective function. Support vector regression [15] has the following modified object function:

$$\min_{W,b,\xi,\xi^*} \frac{1}{2} W^T W + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \tag{2}$$

$$\begin{aligned} \text{subject to} \quad & y_i - (W^T \phi(X_i) + b) \leq \epsilon + \xi_i, \\ & (W^T \phi(X_i) + b) - y_i \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l. \end{aligned}$$

ξ_i is the upper training error (ξ_i^* is the lower training error) subject to the $\epsilon-$insensitive tube $|y - (W^T \phi(X) + b)| \leq \epsilon$ and ϵ is a threshold. C is the cost of error. The cost function ignores any training data that is within the threshold ϵ to the model prediction. This soft margin method increases the robustness of SVR. In the above objective function, $\frac{1}{2} W^T W$ is a regularization term to smooth the function $W^T \phi(X_i) + b$ in order to limit overfitting. Effectively, within the $\epsilon-$insensitive tube, the regularization term constrains the line to be as flat as possible. This flatness is measured by the norm $W^T W$. The parameters of the regression quality are the cost of error C, the width of the tube ϵ, and the mapping function ϕ. Similar to support vector classification, w is a high dimensional vector because ϕ maps data to a higher dimensional space, thus, the dual problem is solved instead:

$$\min_{\alpha,\alpha^*} \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \epsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i (\alpha - \alpha^*) \tag{3}$$

$$\text{subject to} \sum_{i=1}^{l} (\alpha - \alpha^*) = 0, 0 \leq \alpha, \alpha^* \leq C, i = 1, \ldots, l$$

where $Q_{ij} = K(X_i, X_j) \equiv \phi(X_i)^T \phi(X_j)$, $K(X_i, X_j)$ is the kernel function. Commonly used kernel functions are linear, polynomial, Gaussian, sigmoid etc. The derivation of the dual is the same as in support vector classification. The primal-dual relation shows that

$$w = \sum_{i=1}^{l} (-\alpha_i + \alpha_i^*) \phi(X_i) \tag{4}$$

so the approximate function is

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

Fig. 2 shows the architecture of a regression machine constructed by the SV algorithm. It is similar to neural network regression. The difference is that the input layer in SVR are a subset of the training patterns (support vectors) and the test pattern. Fig. 3 shows the architecture of a traditional linear regression where the input layer is the test pattern alone. The complexity of SVR is $O(nl)$, where n is the number of pixels in the input image and l is the number of SVs. The complexity of traditional linear regression is $O(nm)$, where m is the number of coefficients (the size of the patch). In the example shown in Fig. 3, m is 9. SVR has much higher complexity since $l \gg m$. Another difference is that SVR is adaptive while traditional regression is fixed. As in Fig. 2, $(-\alpha_i + \alpha_i^*)$ can be interpreted as the "weight" for each X_i and $K(X_i, X)$ can be viewed as the contribution of each SV. Though the weight is fixed, the contribution of each SV is adaptive to the input pattern X. When X is closer to a SV, that is, when the SV is a good example for the input pattern, then that SV contributes more to the final output. On the other hand, if a SV is different from X, then that SV has less influence in determining the output as $K(X_i, X)$ will be small. Effectively, the kernel function $K(X_i, X)$ measures the distance between a test pattern and the support vector.

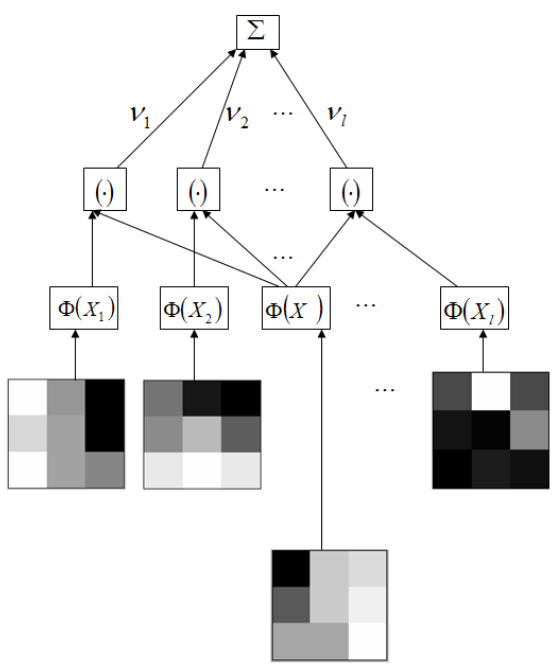

Fig. 2: Architecture of a regression machine constructed by the SV algorithm, $\nu_i = -\alpha_i + \alpha_i^*$, b is added to the sum.

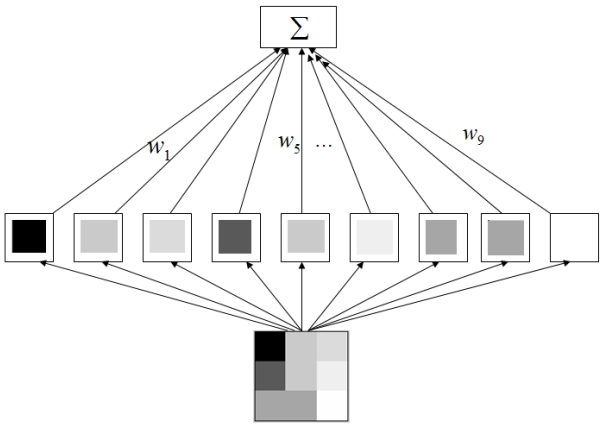

Fig. 3: Linear regression diagram. The intercept b is added to the sum.

3. Experimental Results on Super-resolution

The training set is created by simulating the degradation process that super-resolution method tries to "undo". Here it is assumed that the super-resolution method is to double the size of an image. A high-resolution image is blurred and downsampled to create a low-resolution image that has half the number of original pixels in each dimension. An initial interpolation, such as bicubic interpolation, is applied to the low-resolution image to generate an image of the desired size but the high-resolution details (high-frequencies) are missed. In the training set, we only store the high frequency components of the image (the differences between the image's bicubic interpolation and the original high-resolution image).

In the training set, input is formed by taking a certain size neighborhood from the image g and then converting it into a vector. The output y_i corresponding to this input vector is the central pixel value in the high-frequency image $f(u, v)$. By sliding this neighborhood over all positions on the interpolated image $g(u, v)$, we obtain the training set for SVR. To super-resolve an image, the same size neighborhood window is used to create the input vector, and the pixel value in the high-frequency image is predicted by the trained model obtained in the training step. The image is super-resolved on a pixel-by-pixel basis. LibSVM [16], an implementation of SVR, is used in our experiments. To demonstrate the generalization ability of SVR, the training set is intentionally limited to very few (i.e. 1-3) images in the experiments. However, this does not mean that there are only 1-3 training vectors in the training set. Since the training set is made up of the patches from the training image, a single image might produce a large size training set. The size of training set is determined by the size of image and the size of patch. The test images are selected from several different catalogs in the USC Image Database. In all the experiments, the size of the neighborhood window is 7×7. The default LibSVM SVR parameters are used: Gaussian kernel, $\epsilon = 0.1$ etc.

Supervised machine-learning approaches process test images with the models learned from the training images where the ground-truths are known. Therefore, the approaches are expected to correctly predict the missing high frequency details. The methods attempt

to "restore" rather than "enhance" the image. Given that, PSNR and ISNR are used to objectively measure the quality of the super-resolved (restored) images. PSNR is defined as:

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

where $\hat{f}(i,j)$ is the super-resolved image, and $f(i,j)$ is the original high-resolution image. The size of the images are $M \times N$. From PSNR, ISNR (improvement in signal-to-noise ratio) can be computed as $ISNR = PSNR(\hat{f}) - PSNR(g)$ where g is the interpolated image. Thus, ISNR can reflect the improvement in terms of signal-to-noise ratio.

Some representative images from the USC-SIPI database were tested using the Cameraman SVR model and also the CameramanLena model, which used both the Cameraman and Lena images for training. The images are the 1.1.03 (texture image), 2.1.05 (aerial image), boat, baboon and house. Table 2 shows the results on those images. The performance of the CameramanLena model is better than that for the Cameraman model. This illustrates that, not surprisingly, enlarging the training set can generally improve the performance, as more examples are provided. Experimental result on the Baboon image are shown in Fig. 4. Fig. 4(a) shows the interpolated image and Fig. 4(b) is the Lena SVR super-resolved image. There are more details in Fig. 4(b) than in Fig. 4(a), such as the beard in the lower right, which looks more sharp.

Table 2: PSNR (db) comparison of the interpolated images and the images super-resolved by SVR 1 trained on the Cameraman and SVR 2 trained on the Cameraman and Lena.

Image	interpolated	SVR1	SVR2
texture	22.50	22.80	23.01
boat	28.15	28.53	28.75
baboon	25.48	26.10	26.18
house	31.69	31.79	32.13
aerial	26.94	26.87	27.10

In all the following experiments, the proposed SVR is trained on 3 images. Ni [13] uses 18 images, taken from the CalPhotos collection [17]. The test images are mainly chosen from the USC Image Database. Additionally, some document images are used. The size of the neighborhood window is 7×7 in our SVR method. Table 3 summarizes the comparative results of the two SVR methods and the kernel regression method. In the result provided by Ni, 6 out of the 8 test images have negative ISNR, that is, in terms of PSNR, the super-resolved images are worse. Our result indicates that all the super-resolved images are closer to the ground-truth images. Though both methods are based on SVR, the two methods have differences regarding how the input/output are formulated and how SVR is implemented. We use one SVR that maps patch to single pixel and Ni uses N by N SVRs that maps a patch to N by N pixels ($N = 2$ if we double the size of image). While we use a public available SVM software package, Ni [13] enhanced the SVR problem first by finding the optimal kernel and then implemented their own SVR. The above differences lead to the performance differences. In terms of PSNR, the kernel regression method provides the worst result. Fig. 7 shows the up-scaled House image. It is obvious that the image is smoother than the interpolated image, e.g., the texture on the wall are largely smoothed out.

(a)

(b)

Fig. 4: (a) The interpolated Baboon image. (b) The SVR super-resolved image.

Results on the document image are reported in Fig. 5. PSNR of the interpolated image is 20.64 dB. The PSNR and ISNR of the super-resolved image by the proposed method are 21.85 dB and 1.21 dB, respectively. The PSNR of the one processed by [13] is 20.49 dB, which is lower than those of both the proposed one and the interpolated one. In terms of ISNR (-0.15 dB), the super-resolved image is slightly worse, though the image is visually sharper. This indicates that Ni's method acts more like a "sharpening" operation. One can observe some artifacts in the image, especially around the texts in the image. To better illustrate the difference, Fig. 5 shows the zoomed-in regions. The result of the proposed method does not have those artifacts. Fig. 6 shows the result on the House image. The same artifices are observed in Ni's result.

Table 3: PSNR (dB),ISNR comparisons of our method (3 training images),Ni[13] (18 training images) and kernel regression.

Image	Interpolated	Proposed	Ni	Kernel Regression
baboon	21.3601	0.5255	-0.0394	-1.1767
boat	25.7552	0.3872	-0.0671	-2.0542
cameraman	26.3003	0.1906	0.2768	-1.9424
doc	20.6429	1.2073	-0.1497	-1.8689
house	30.8212	1.9985	0.0716	-3.2937
lena	28.8343	0.1483	-0.4037	-2.8143
peppers	29.5135	0.3231	-1.1959	-3.4137
tree	27.0343	1.4069	-0.6657	-4.037

4. Conclusion

All example-based methods learn from examples. Some method tries to explicitly build models such as Markov network and others might use simpler learning such as nearest neighborhood. In our case, we use SVR. SVR relies on SVs and their associated ν_i in predicting the output image pixels. Since SVR can generalize well to unseen data, even trained on images from different classes, SVR can super-resolve a variety of images. In other words, the local similarity assumption is relaxed by SVR in practise.

Simulations show that SVR is able to identify a set of generally applicable SVs from a very small image data set for super-resolution. The learned model has been tested on a variety of different types of images (texture, aerial, nature, document). Although the PSNR improvement is not dramatic, visual inspection suggests that some missed high-frequency component are recovered. Experiments suggest that the performance of SVR is improved when trained on a larger data set, as with other machine learning algorithms. When the training images match well with the test images, a better model is generated that predicts the missed high-frequency information more accurately. This indicates that it helps to have an image classifier before super-resolution. Comparative results indicate that the proposed method outperforms a competing SVR based super-resolution in terms of PSNR with fewer images for training and less number of SVRs.

References

[1] A. M. Tekalp, *Digital Video Processing*, Prentice-Hall, Englewood Cliffs, 1995.
[2] R. S. Wagner, D. E. Waagen, and M. L. Cassabaum, "Image superresolution for improved automatic target recognition," in *Proc. SPIE*, 2004, vol. 5426, pp. 188–196.

[3] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 277–285, March 1991.

[4] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, Feb. 2007.

[5] C. V. Jiji and S. Chaudhuri, "Single-frame image super-resolution through contourlet learning," *EURASIP J. Appl. Signal Process.*, vol. 2006, no. 1, pp. 235–235, uary.

[6] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graphics and Applicat.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.

[7] M. Elad and D. Datsenko, "Example-based regularization deployed to super-resolution reconstruction of a single image," *The Computer Journal*, vol. 50, no. 4, Apr. 2007.

[8] D. Datsenko and M. Elad, "Example-based single image super-resolution: A global map approach with outlier rejection," *Journal of Multidimensional System and Signal Processing*, vol. 18, no. 2-3, Sept. 2007.

[9] C. V. Jiji, S. Chaudhuri, and P. Chatterjee, "Super-resolution: Should we process locally or globally?," *Multidimensional Systems & Signal Processing*, vol. 18, no. 2-3, Sept. 2007.

[10] D. Li, R. Mersereau, and S. Simske, "Blind image deconvolution through support vector regression," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 931–935, May 2007.

[11] D. Li, R. M. Mersereau, and S. Simske, "Blind image deconvolution using support vector regression," in *Proc. IEEE. Int. Conf. Acoustics, Speech, Signal Process.*, Philadelphia, USA, 2005, vol. 2, pp. 113–116.

[12] D. Li, R. M. Mersereau, and S. Simske, "Image denoising through support vector regression," in *Proc. IEEE Int. Conf. Image Process.*, San Antonio, USA, 2007.

[13] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, vol. 16, pp. 1596–1610, June 2007.

[14] *University of Southern California - Signal & Image Processing Institute - The USC-SIPI Image Database*, http://sipi.usc.edu/services/database/Database.html.

[15] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, 1995.

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

[17] *Univ. of Calif. Berkeley - The Biodiversity Sciences Technology (BSCIT) group -CalPhotos*, http://calphotos.berkeley.edu/index.shtml.

Fig. 5: Results on the Doc image. Upper left: the ground-truth image. Upper right: the interpolated low-resolution image, PSNR=20.64 dB. Middle left: super-resolved by the proposed method, PSNR=21.85 dB. Middle right: the result of Ni, PSNR=20.49 dB. Lower left: a zoomed-in region of the proposed method. Lower right: a zoomed-in region of result by Ni.

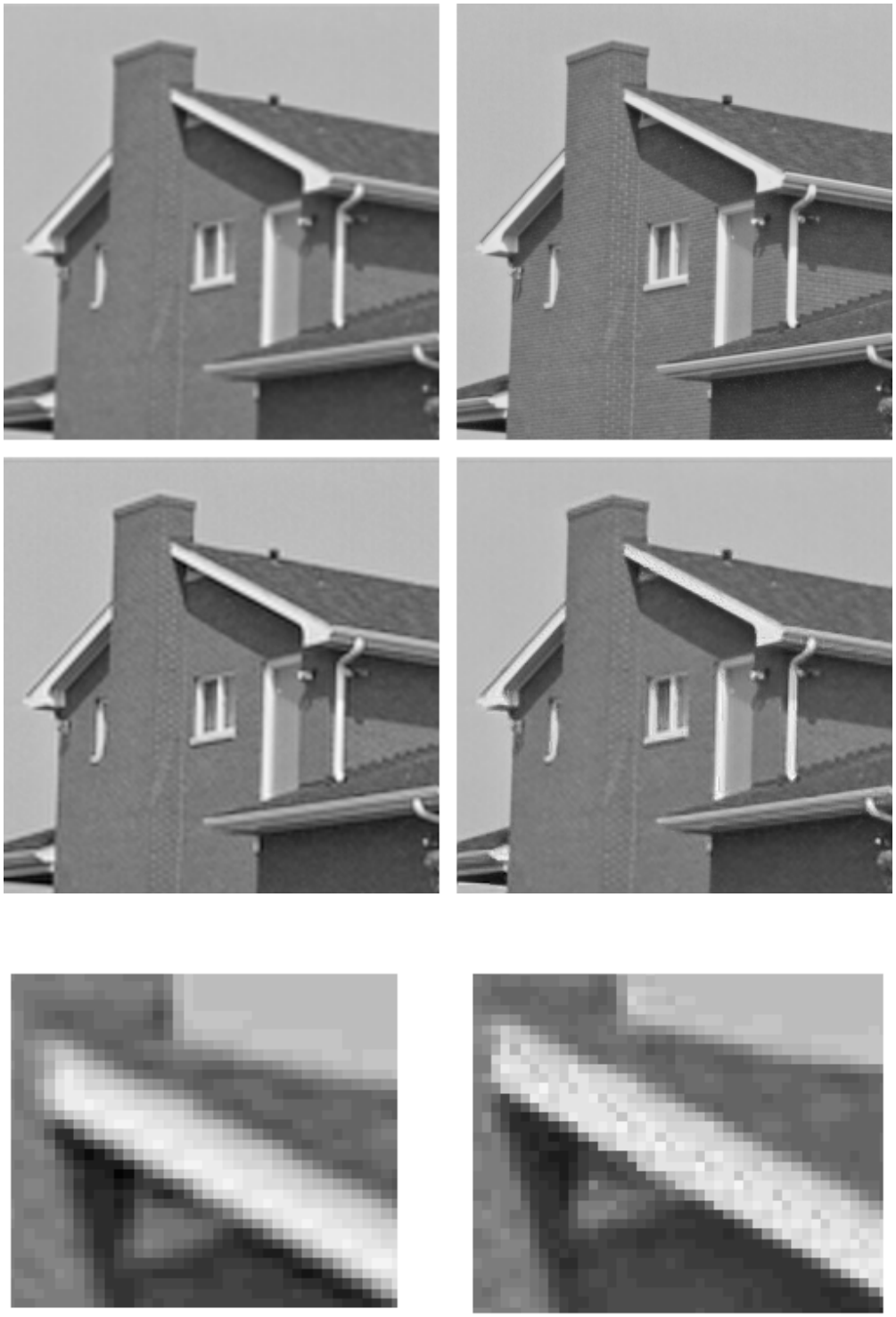

Fig. 6: Results on the HOUSE image. Upper left: the interpolated low-resolution image, PSNR=30.82 dB. Upper right: the ground-truth image. Middle left: super-resolved by the proposed method, PSNR=32.82 dB. Middle right: the result of Ni, PSNR=30.89 dB. Lower left: a zoomed-in region of the proposed method. Lower right: a zoomed-in region of result by Ni.

Fig. 7: The House image upscaled by the kernel regression method.