

Image super-resolution representation via image patches based on extreme learning machine

Qiuxi Zhu, Xiaodong Li, Weijie Mao
 Department of Control Science and Engineering
 Zhejiang University
 Hangzhou, China
 e-mail: zhuqiuxi0743@126.com

Abstract—In this paper, aimed at the extensively existing problem of slowness in mainstream image super-resolutions, an efficient approach is proposed for super-resolution based on the extreme learning machine (ELM) for single-hidden layer feedforward neural networks (SLFNs). Features and issues (e.g. parameter selections) in the application of ELM are discussed, on the basis of which a general framework for a variety of super-resolution problems is proposed, and corresponding experiments are conducted. It is shown in the results that the proposed approach can achieve relatively good quality and much faster speed compared to traditional reconstruction-based super-resolutions, therefore the effectiveness of this method is demonstrated.

Keywords—ELM; neural network; image processing; super-resolution

I. INTRODUCTION

Image super-resolution is a significant branch in image fusion, which rebuilds high-resolution (HR) images by utilizing information provided in low-resolution (LR) images or sequences of correlated LR images [1]. Due to inevitable limits of hardware conditions, complicated structures and high expenses, acquirement of images that are of enough resolution is quite a hot topic in many related fields like medical diagnoses, monitoring, military investigations and remote mapping, to which image super-resolution might be a cheap but effective solution.

It has been more than three decades since the concept of image super-resolution was proposed. Three phases in its development is witnessed: interpolation, reconstruction and learning-based super-resolution. Extensive works have been done on interpolations and reconstructions, thus the development of these methods was found sluggish in recent years[2]. Interpolations are fast but usually fail to acquire good quality in rebuilt images and to deal with multi-frame super-resolution problems. Reconstructions like projection onto convex sets (POCS) method [2] and maximum posterior probability (MAP) method [4-7] are based on empirical approximations and iterative refinements which sometimes end up with relatively high-quality rebuilt images but the iterations always lead to slow computations which set inevitable limits in the application of these methods.

In recent years, learning-based super-resolutions are drawing increasing attention [8,9], providing us with the possibility of utilizing priori knowledge given in training sets

and make it possible to increase the resolution of LR images by more than 3 or 4 times [2]. However, most of the applied well-developed learning algorithms such as back propagation (BP) and support vector machine (SVM), are still trapped by the bottleneck of slowness caused by iterative solutions and thus fail to be accepted in fields that lay stress on the speed of imaging [2].

In this paper, extreme learning machine (ELM) [10] for single hidden-layer feedforward neural networks (SLFNs) is applied to image super-resolution in order to provide learning-based super-resolutions with a fast and efficient learning algorithm. The proposed method mainly focuses on providing the super-resolution problem with a general framework (including the configurations of the neural network) that can be applied to different types of super-resolution problems with few modifications needed in transplantations. By training the SLFN with relative training images, super-resolution images can be rebuilt by the integration of network outputs.

The remaining parts of this paper are organized as follows. In Section II, the algorithm of ELM is briefly introduced. The technical approach to implement the proposed method is described in Section III and the experimental sets and results are shown in Section IV, as well as comparisons to traditional interpolations and image reconstruction using fast image inpainting algorithms, a late research in reconstruction-based methods [12]. And finally in Section V, this paper is concluded and a few of interesting topics are proposed for future works.

II. EXTREME LEARNING MACHINE

Extreme learning machine (ELM) [10,11] was proposed by Huang et al. as an efficient learning algorithm for single hidden-layer feedforward neural networks (SLFNs). A typical SLFN is mathematically modeled as follows

$$\mathbf{o} = \mathbf{o}(x) = \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x} + b_i), \quad (1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbf{R}^n$ is the input vector of SLFN, $\mathbf{o} = [o_1, o_2, \dots, o_m]^T \in \mathbf{R}^m$ is the output vector, $\tilde{N} \in \mathbf{N}_+$ is the count of hidden nodes, $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T \in \mathbf{R}^n$ is the input weight vector of the i -th hidden node, $b_i \in \mathbf{R}$ is the bias

of the i -th hidden node, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T \in \mathbf{R}^m$ is the output weight vector of the i -th hidden node, $g: \mathbf{R} \rightarrow \mathbf{R}$ is the activation function of shared by all the hidden nodes, $\mathbf{w}_i \cdot \mathbf{x}$ stands for the inner product of \mathbf{w}_i and \mathbf{x} .

Given the training set

$$\mathcal{S} = \left\{ (\mathbf{x}_j, \mathbf{t}_j) \mid \mathbf{x}_j \in \mathbf{R}^n, \mathbf{t}_j \in \mathbf{R}^m, j = 1, 2, \dots, N \right\},$$

where there are N training samples, ELM randomly chooses \mathbf{w}_i , the input weight vectors, and b_i , the hidden layer biases, and attempts to solve the following linear system to acquire the minimum training error as well as the smallest norm of the solution [10]

$$\mathbf{H}\beta = \mathbf{T}, \quad (2)$$

where \mathbf{H} is the hidden layer output matrix that is defined as

$$\mathbf{H} = \begin{pmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{pmatrix}_{N \times \tilde{N}}, \quad (3)$$

and the training output matrix \mathbf{T} and the output weight matrix β are defined as

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}.$$

The minimal norm least square solution to the equation above is [13]

$$\hat{\beta} = \mathbf{H}^\dagger \mathbf{T}, \quad (4)$$

where \mathbf{H}^\dagger is the Moore-Penrose generalized inverse (or pseudo inverse) of the hidden layer output matrix \mathbf{H} . $\hat{\beta}$, the output weight matrix provides the SLFN with the minimum training error and the smallest norm of the output weights.

ELM overwhelms the traditional learning algorithms by avoiding the time-consuming iterative parameter tuning. It is extremely fast and usually ends up with better generalization performance than iterative algorithms like BP.

III. TECHNICAL APPROACH

The technical approach of the proposed method is shown in Fig. 1.

In the training phase, sequences of LR images are generated from HR images by under-sampling using the imaging model. An LR sequence consists of one or more LR frames. Then, the LR sequences and the corresponding HR images are turned into

training input vectors and output vectors so that ELM algorithm could be applied and the parameters of the SLFN are determined.

In real applications, the ELM trained SLFN can be used for super-resolution of LR sequences whose acquisition system can be modeled as the imaging model used in the training phase.

In the testing phase, in order to acquire the performance indexes for future comparisons, the LR sequences are once more generated from the testing HR images using the imaging model, and used for the generation of testing input vectors. The testing inputs are processed by the ELM trained SLFN to acquire the testing outputs, which are later used for the reconstruction of the rebuilt HR images. Finally, the rebuilt images are compared to the original HR images for the calculation of performance indexes such as root-mean-square error (RMSE) and peak signal-to-noise ratio (PSNR).

The proposed approach could be seen as a general framework for super-resolution problems that can be applied to different types of super-resolutions (independent from the problem details such as the selection of imaging model and the count of frames in LR sequences) and even some other image fusions, with few modifications needed in transplantations.

In the remaining parts of this section, details in the proposed technical approach is discussed, including the configurations of the neural network, the acquisition of network input and output vectors from the corresponding LR sequences and HR images, the reconstruction of the rebuilt HR images from the output vectors, and the performance indexes used in the comparisons.

A. Network Configurations

The configurations of the neural network refer to the structure and parameters of the SLFN which is to be trained by the ELM algorithm. According to the ELM algorithm, some configurations should be independent from the specific problem and always stay the same, some configurations depend on the problem description, and some configurations are determined in the training phase by ELM.

In the proposed approach, the following configurations of the SLFN are designed to be problem-independent and are fixed before the execution of the algorithm.

- All input nodes share the linear activation function $f: \mathbf{R} \rightarrow \mathbf{R}$, $f(x) = x$.
- All input nodes have unit input weights and zero biases.
- All hidden nodes share the sigmoid activation function

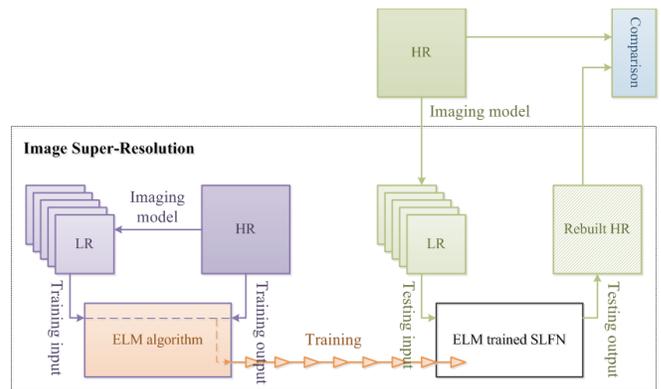


Figure 1. Technical approach used in this paper.

$g : \mathbf{R} \rightarrow \mathbf{R}$,

$$g(x) = \frac{1}{1 + \exp(-x)}.$$

- All output nodes share the linear activation function $f : \mathbf{R} \rightarrow \mathbf{R}$, $f(x) = x$.
- All output nodes have zero biases.

The following configurations are problem-dependent and should be determined or calculated before the execution of ELM algorithm.

- The input dimension of the SLFN, n , is calculated in the following expression.

$$n = N_x (s^2 + 5), \quad (5)$$

where N_x is the count of frames in an LR sequence, s is the size of LR image blocks.

- The output dimension of the SLFN, m , is calculated in the following expression.

$$m = s^2 L^2, \quad (6)$$

where L is length resizing ratio of the imaging model, for example, if an HR image in resolution 320×240 is down-sampled to a sequence of LR images that are all in resolution 160×120 , then there is $L=2$ for this LR imaging model.

- The count of hidden nodes of the SLFN, \tilde{N} , should be determined by structure learning based on the problem and the characteristics of the training set, which is not covered in this paper. So in this paper, different values are iteratively tested and an optimal one is used in further comparisons

Finally, the following configurations are determined during the execution of the algorithm.

- The input weights w_i and the hidden layer biases b_i are randomly chosen.
- The output weights described as the output weight matrix β is computed by (4)

$$\beta = \mathbf{H}^\dagger \mathbf{T}. \quad (7)$$

B. Image Vectorization

SLFNs discussed above are based on the structure of vector mapping networks, where the inputs and outputs of the neural networks are real vectors. Therefore, LR sequences and HR images need to be converted into input and output vectors respectively before ELM could be applied.

Vector mapping pairs are based on image block mapping groups. An image block mapping group consists of N_x LR image blocks of size $s \times s$ taken respectively from the LR frames in an LR sequence, and one HR image block of size $sL \times sL$ taken from the HR image corresponding to the LR sequence. All LR blocks should share the coordinates on the original LR frames where they are taken, and the HR block should just cover the corresponding region where the LR blocks are down-sampled from.

Then all these image blocks, represented in pixel value matrices, are “stretched” into column vectors. Such an image block vector contains all pixel values from the corresponding image block matrix line by line. The LR block vectors have length s^2 and the HR block vectors have length $s^2 L^2$.

In order to take the directional information in the training images into consideration, in addition to the s^2 pixel values in the LR block vectors, 1st and 2nd order derivatives [14] described below at the center of the LR image blocks are appended to the LR vectors

$$\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial y^2}, \frac{\partial^2 I}{\partial xy}.$$

Thus, LR vectors turn to have length (s^2+5) . Finally, all the LR vectors in a single group are connected end to end, forming a single input vector of length $N_x(s^2+5)$. The HR block vector in the group is treated as the output vector of length $s^2 L^2$, and together with the input vector, it forms a vector mapping pair corresponding to the chosen group.

Repeated iteratively for all possible image block mapping groups, the steps above provide with all potential vector mapping pairs in the given training images. For pixels outside the border of the image, values are determined by the reflective boundary conditions.

In testing cases, the rebuilt HR vectors are calculated by the ELM trained SLFN based on the testing inputs and used for the reconstruction of the rebuilt HR images. Comparisons are made directly to the original HR images.

C. Image Reconstruction From Output Vectors

The reconstruction of rebuilt HR image from the output vectors of the ELM trained SLFN could be considered the inverse of the image vectorization described in III.B. However, there is an explicit problem that must be dealt with elegantly in this step.

To utilize all the information sufficiently in the testing phase, input vectors might be converted from LR blocks that have some rectangle regions of pixels overlapped. Consequently, it is possible that a single pixel on a rebuilt HR image is described differently in more than one output vectors.

To overcome this ambiguity, in the testing phase of the proposed approach, coordinates of the LR blocks on the original LR frames are recorded and therefore the coordinates of the corresponding HR block on the rebuilt HR image could be computed. If a pixel has multiple definitions in more than one output vectors, the final pixel value is determined by a Gaussian kernel based on all these definitions.

D. Performance Indexes

In this paper, the following performance indexes, which are widely used in image super-resolution researches as well as other image processing topics, are used in the tests for comparisons.

1) Running time

For learning-based methods, running time includes training time and testing time.

In the proposed approach, training time is defined as the total time of the vectorization of images and the running time of the ELM algorithm. It is mostly spent on the calculation of the hidden layer output matrix and its pseudo inverse. The complexity of these steps is mainly contributed to by the size of the training set.

In real applications, the testing phase of the proposed method is expected to be executed much more times than the training phase. Therefore, testing time is also considered as a significant performance index and will be compared to the running time of the traditional super-resolutions in Section IV.

Testing time is defined as the total time of the calculation of the output vectors using the ELM trained SLFN and the reconstruction of rebuilt HR image from the output vectors. The time used for the reconstruction of a rebuilt HR image depends only on the size of the HR image, thus the testing time-to-count of hidden nodes curve is expected to have a positive intercept on the vertical axis and thus testing time may exceed training time for some cases.

2) Root mean square error (RMSE)

For applications in image processing, RMSE between the rebuilt image and the original image is defined as

$$RMSE(\mathbf{O}_{H \times W}, \mathbf{T}_{H \times W}) = \sqrt{\frac{\sum_{i=1}^H \sum_{j=1}^W (\mathbf{O}[i, j] - \mathbf{T}[i, j])^2}{HW}}, \quad (8)$$

where \mathbf{O} is the pixel value matrix of the rebuilt HR image, \mathbf{T} is the pixel value matrix of the original HR image, $\mathbf{O}[i, j]$ stands for the pixel value in the i -th row and in the j -th column of \mathbf{O} , and $H \times W$ is the resolution of these HR images.

3) Peak signal-to-noise ratio (PSNR)

Peak signal-to-noise ratio (PSNR) is a very common term and index used in signal processing engineering. It is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is usually measured in decibel (dB).

PSNR is defined as follows

$$PSNR(\mathbf{O}, \mathbf{T}) = 20 \log_{10} \frac{MAX_I^2}{RMSE(\mathbf{O}, \mathbf{T})}. \quad (9)$$

In our representation, $MAX_I = 1$. The expression above shows the relationship between PSNR and RMSE. In tests, once RMSE is calculated and MAX_I is already known, PSNR can be uniquely determined.

IV. EXPERIMENTS

The programs were written in C++ and the executable binaries were compiled under Microsoft Windows 7 (32-bit) Service Pack 1 using MinGW G++ (GCC) 4.7.0. The programs depend on OpenCV 2.4.3 as the image processing library and LAPACK 3.4.2, OpenBLAS develop and Armadillo 3.810.1 as linear algebra libraries. Tests were executed on a desktop computer with a DualCore AMD Athlon 64 X2, 2600 MHz (13 x 200) 5000+ CPU and 3.00 GB of RAM.

Both the single-frame (in which there's only one LR frame in an LR sequence) cases and the multi-frame cases were tested. In single-frame tests, the effect that the chosen of \tilde{N} has on the speed and quality of the proposed algorithm was tested. The results were compared to traditional interpolations such as bilinear and bicubic interpolation. In multi-frame tests, results were compared to that of image reconstruction using fast image inpainting algorithms [12], a late research in reconstructions.

To reveal the effect that the chosen of \tilde{N} has on the speed and quality, the 256×256 Lena grayscale picture was used as the HR training image and the Woman 1 grayscale picture of the same size were used as the HR testing images. LR images were generated using the pyramid-down provided by the OpenCV library. In the experiment, $s=3$ and cases where $L=2$ and $L=4$ were tested respectively. Each data point reflects the average of 20 repeats.

Effect the chosen of \tilde{N} has on the speed of the algorithm when $L=2$ is shown in Fig. 2. Similar curves are produced for the case of $L=4$. A positive vertical intercept is found on the testing time-to-count of hidden nodes curve, which has noted in III.D.

Effect the chosen of \tilde{N} has on the quality when $L=2$ is shown in Fig. 3. It can be found that the testing PSNR increases as the count of hidden nodes and reaches its maxima where the count of hidden nodes is about 50. Similar curves are resulted for the case of $L=4$.

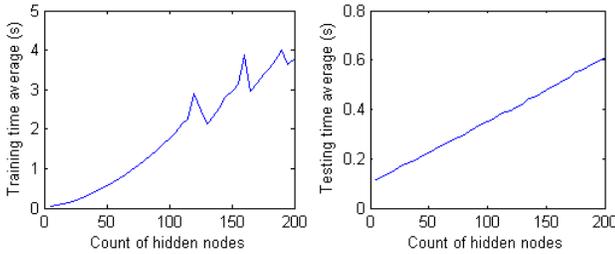


Figure 2. Effect that the chosen of the count of hidden nodes has on the training (left) and the testing (right) time, $L=2$.

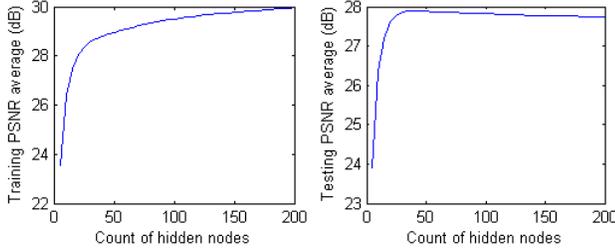


Figure 3. Effect that the chosen of the count of hidden nodes has on the training (left) and the testing (right) PSNR, $L=2$.

These figures show that the increasing of the count of hidden nodes doesn't always lead to the improvement of the quality of rebuilt images. A proper count of hidden nodes could provide the algorithm with both speed and quality. Experimental data of the tests described above is in Table I and Table II for references.

In the comparisons with traditional interpolations, face images from the Georgia Tech face database were used as training and testing samples. The size of the HR face images were about 300×300 . In the experiment, $s=3$ and $L=3$. For each testing image, the result was the average of 20 repeats. It can be seen in the image comparisons in Fig. 4 that more details are acquired by the proposed method especially near the edges than by bicubic interpolation. Experimental data of the comparisons is in Table III for references.

In the comparisons with image reconstruction using fast image inpainting algorithms [12], the testing image was the classic 126×126 Camera picture used in [12]. In the experiment, $s=2$ and $L=3$. In these comparisons, multi-frame super-resolutions are encountered, which are described as "multi frame, no blur" and "single frame, multi blur" respectively in [12]. The results of the proposed method were average values of 20 repeats respectively. Experimental data of the comparisons is in Table IV for references.

Rebuilt images are compared in Fig. 5. The rebuilt images by the proposed method seems to have more details around the edges, when the PSNR quality of the proposed method failed to overwhelm the image reconstruction using fast image inpainting algorithms [12], because of the recognizable grid effect in the rebuilt images, to which the lack of necessary priori information in the training images might have contributed.

However, the proposed method can be hundreds of times faster than the reconstruction-based method, as is expected, even when the training time is included in the total running time of the proposed method indicating that the learning process is repeated every time the proposed method is applied

TABLE I. TRAINING TIME AND TESTING TIME FOR DIFFERENT CHOSEN COUNTS OF HIDDEN NODES, $L=2$

\bar{N}	Training			Testing		
	Time/s	RMSE	PSNR/dB	Time/s	RMSE	PSNR/dB
30	0.267	0.0372	28.58	0.177	0.0404	27.86
35	0.336	0.0368	28.69	0.190	0.0403	27.89
40	0.407	0.0364	28.78	0.201	0.0404	27.88
45	0.490	0.0359	28.89	0.213	0.0403	27.89
50	0.577	0.0356	28.96	0.225	0.0403	27.89
55	0.656	0.0354	29.03	0.239	0.0404	27.88
60	0.756	0.0351	29.11	0.251	0.0404	27.88
65	0.853	0.0348	29.16	0.263	0.0405	27.86
70	0.966	0.0346	29.22	0.276	0.0405	27.85

TABLE II. TRAINING TIME AND TESTING TIME FOR DIFFERENT CHOSEN COUNTS OF HIDDEN NODES, $L=4$.

\bar{N}	Training			Testing		
	Time/s	RMSE	PSNR/dB	Time/s	RMSE	PSNR/dB
10	0.023	0.0707	23.02	0.103	0.0671	23.47
15	0.032	0.0664	23.55	0.106	0.0635	23.95
20	0.041	0.0638	23.91	0.109	0.0617	24.20
25	0.052	0.0626	24.07	0.113	0.0611	24.27
30	0.063	0.0616	24.21	0.118	0.0607	24.34
35	0.075	0.0611	24.28	0.122	0.0607	24.34
40	0.090	0.0605	24.36	0.126	0.0606	24.36
45	0.110	0.0601	24.43	0.130	0.0605	24.36
50	0.132	0.0596	24.49	0.134	0.0605	24.37

TABLE III. COMPARISONS BETWEEN TRADITIONAL INTERPOLATIONS AND THE PROPOSED METHOD.

Testing image	Traditional interpolations			Proposed method
	Nearest neighbor	Bilinear	Bicubic	
s01/05	29.94	32.13	32.06	32.74
s08/01	26.68	28.28	28.28	28.61
s08/03	27.55	29.57	29.54	30.01
s10/01	26.44	28.42	28.34	28.90
s10/02	26.12	28.16	28.10	28.59
s23/01	28.33	30.18	30.08	30.57
s23/13	27.42	29.28	29.17	29.72

TABLE IV. COMPARISONS BETWEEN TRADITIONAL INTERPOLATIONS AND THE PROPOSED METHOD.

Test case	Method	Training time/s	Testing		
			Time/s	RMSE	PSNR/dB
Multi frame, no blur	Fast inpainting [12]	-	21.67	-	25.10
	Proposed	1.648	0.075	0.0547	25.24
Single frame, multi blur	Fast inpainting [12]	-	110.5	-	22.82
	Proposed	0.645	0.046	0.0865	21.26

which is seldom seen in real applications. This case demonstrated the eligibility of the proposed method to acquire much faster speed of super-resolution compared with advanced

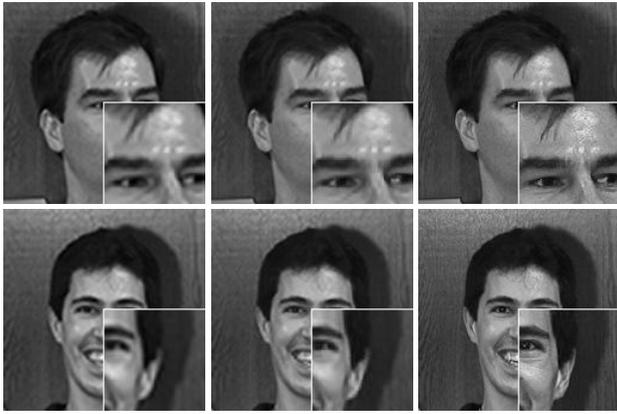


Figure 4. Comparisons: bicubic (left), the proposed method (middle), and original images (right); Image s01/05 (first line, 185 hidden nodes) and image s23/13 (second line, 115 hidden nodes).



Figure 5. Comparisons: image reconstruction using fast image inpainting [12] (left), the proposed method (middle) and the original images (right); Multi frame, no blur (first line, 215 hidden nodes for the proposed method) and Single frame, multi blur (second line, 105 hidden nodes).

reconstruction-based methods with the quality of rebuilt images basically not influenced.

V. CONCLUSIONS

In this paper, ELM for SLFNs is applied to the problem of image super-resolution. A general framework for multiple varieties of image super-resolutions problems is proposed, in which few modifications are expected in transplantations.

Experiments are conducted and the results are compared respectively with traditional interpolations and the image super-resolution using fast image inpainting algorithms [12]. The advantages of the proposed method, especially those of the speed are revealed in comparisons, and therefore a bright future of learning-based image super-resolutions is demonstrated. Better quality of rebuilt images is expected if a comprehensive set of training images of a specific field could be introduced with required features contained.

Effect the different chosen counts of hidden nodes have on the speed and quality of the process of image super-resolution is analyzed based on the experimental data of the single-frame cases. It is indicated that there usually exists a optimal count of hidden nodes in a specific problem. The determination of the proper count of hidden nodes and the chosen of proper training images are quite interested topics which require further considerations.

ACKNOWLEDGMENT

This work was supported by the Zhejiang Provincial Natural Science Foundation of China (No.LR12F03002) and the National Natural Science Foundation of China (No.61074045).

Corresponding author e-mail: wjmao@iipc.zju.edu.cn

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, pp.21-36, 2003.
- [2] Jian Pu, Jun-ping Zhang, and Hua Huang, "A survey of super resolution algorithms," *Journal of Shandong University (Engineering Science)*, vol. 39, pp.27-32, 2009.
- [3] Henry Stark, and Peyma Oskoui, "High-resolution image recovery from image-plane arrays, using convex projections," *J. Opt. Soc. Am. A*, vol. 6, pp.1715-1726, 1989.
- [4] Richard R. Schultz, and Robert L. Stevenson, "Improved definition video frame enhancement," *International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 1995.
- [5] Richard R. Schultz, and Robert L. Stevenson, "Video resolution enhancement," *Proc. SPIE*, vol. 2421-03, 1995.
- [6] Richard R. Schultz, and Robert L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, vol.5, pp.996-1011, 1996.
- [7] Richard R. Schultz, and Robert L. Stevenson, "Motion-compensated scan conversion of interlaced video sequences," *Proc. SPIE*, pp.107-118, 1996.
- [8] Le An, and Bir Bhanu, "Improved image super-resolution by Support Vector Regression." *The 2011 International Joint Conference on Neural Networks (IJCNN)*, 2011.
- [9] Kwang In Kim, and Younghee Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp.1127-1133, 2010.
- [10] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol.70, pp.489-501, 2006.
- [11] Guang-Bin Huang, et al, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, pp.513-529, 2012.
- [12] Tony F. Chan, et al, "Superresolution image reconstruction using fast inpainting algorithms," *Applied and Computational Harmonic Analysis* vol.23, pp.3-24, 2007.
- [13] Denis Serre, *Matrices: Theory and applications*. vol.216. Springer, 2010.
- [14] Hany Farid, and Eero P. Simoncelli, "Differentiation of discrete multi-dimensional signals," *IEEE Transactions on Image Processing*, vol.13, pp.496-508, 2004.
- [15] Jian Sun, Zongben Xu, and Heung-Yeung Shum, "Image super-resolution using gradient profile prior," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.