

# Extreme Learning Machine based Single Image Super Resolution

---

## Abstract

Although Convolution Neural Network(CNN) based methods have recently become popular for image super-resolution, most of them are time-consuming and computationally complex. It is difficult to apply those methods to real-time applications or devices with low compute capability. In this paper, we propose a novel method for fast single image super-resolution. We use a decision tree to divide image patches to different subspaces fast. Patches divided into same subspace suppose to be similar and it is accurate and efficient to learning a mapping function from Low-Resolution(LR) space to corresponding High-resolution(HR) space. Extreme Learning Machine(ELM) is a fast learning algorithm because its special traits. We use ELMs to learn mapping functions for every subspace. All subspaces mapping models and the decision tree form a piece-wise non-linear function. Extensive experiments on benchmark datasets show that the utility of the proposed method for image super-resolution in speed and accuracy.

*Keywords:* Single image super-resolution, Extreme Learning Machine, Decision tree

---

## 5 1. Introduction

Image super-resolution is to reconstruct a high resolution from one or more given low resolution image. Lots of high frequency information such as tiny textures is lost in low-resolution image, which makes SR problem highly ill-posed. Image super-resolution plays an important role in many fields, such as video surveillance, remote sensing and image sensing. The Eq. 1 shows the relationship between low-resolution images and its high-resolution image.

$$X = HY + n \quad (1)$$

where  $H$  is the down-sample kernel, and  $n$  is the noise. The low-resolution image  $X$  is derived from its origin image, which is down-sampled and noised. In general, single image super-resolution methods can be divide into three categories: interpolation-based method, learning-based SR methods and reconstruction-based SR methods.

10     The simplest super-resolution (SR) method is the interpolation-based method. This kind of SR method uses a linear kernel to filter the input LR image to obtain the HR image. The interpolation kernel is single and low-pass, thus it can not estimate the potential image structure. In addition, the information provided by the input LR image is limited, which causes the smoothness in the output HR image. Thus, the interpolation-  
15     based methods can not satisfy the reality needs.

   The learning-based SR methods learn the mapping relationship between the LR space and HR space to address the SR problem. Most existing learning-based methods can classify into two classes: the coding-based methods and the regression-based methods. The coding-based methods assume that there exists isometric manifolds in  
20     the coupled LR-HR space. This assumption allows encoding coefficients in LR space to be directly passed to the corresponding HR space. Then, they are used for SR reconstruction. Lu et al. proposed an SR method based on locally linear embedding [5]. The SR method proposed by Gao et al. combines sparse representation and neighborhood embedding [25]. The sparse-coding-based SR methods construct the corresponding  
25     dictionary from the LR space and the HR space, respectively. The LR dictionary is used to compute the encoding coefficients for the input LR feature. Then, the coefficients can be used directly in the HR dictionary. The target HR output is obtained by computing the weighted sum of the HR dictionary atoms. According to the way the dictionaries are constructed, the sparse-coding-based SR methods can also be clas-  
30     sified into two categories: the orthogonal dictionary based methods [28][9] and the over-complete dictionary based methods [42] [41][26][38]. The regression-based SR methods directly learn the mapping from the LR space to HR space. The SR method proposed by Freeman et al. uses the Markov random field (MRF) [10]. Kim et al addressed the SR problem by sparse representation and kernel regression [21]. Moreover,  
35     there are SR methods based on linear regression [40][44], SVM regression [29], and Gaussian process regression [37][11][23][36]

The reconstruction-based SR methods [18][3] [24] use multiple LR images to reconstruct an HR image. These LR images all come from the same scene, and there must be sub-pixel shifts between them. In real, users may not necessarily obtain multiple LR images that meet this requirement. Thus, its application is limited.

Some researchers train a deep convolution neural network to learn the mapping model from the LR space and HR space [19] [20] [22] [31]. This type of SR methods greatly improve the image SR. However, the training process is complicated, and it involves a longer training time. Based on the obstacles of training neural network and the advantages of extreme learning machine (ELM), we propose an ELM-based single-image SR method. The proposed method employs the piecewise non-linear regression scheme. First, it conducts Hadamard transform on LR training data (LR training image patches) to obtain Hadamard patterns. Then, the training data (LR-HR training image patch pairs) are classified based on the generated Hadamard patterns. Last, each subspace is used to train a mapping model, which is a non-linear mapping model from the LR space to HR space, ELM. All of the mapping models form a piecewise non-linear SR function. Our method also employs the ternary decision tree. Experimental results show that the proposed method yields competitive objective evaluation results and better subjective reconstruction results at the fastest running speed.

The remainder of this paper is organized as follows: Section II introduce the related work of algorithms about image super-resolution. Section III describe the fundamental concepts and theories of ELM. Section II presents the proposed method. Section V gives the experiment results on different datasets and the visual comparisons with other methods are also included. Finally, Section II draws conclusions.

## 2. Related Work

ELM is used for training a single-hidden-layer feed-forward network, and some researchers tried to use it for HR image reconstruction [2] [32] [6]. The ELM-based single-image SR method was first introduced in [2]. This method firstly interpolates the input LR image to the desired size. Then, it extracts the first-order and second-order gradients from the interpolated image as high-frequency features. After that, a

traditional ELM is used to perform the regression operation. The mapping relationship between the LR input and the high-frequency feature is learned. For HR reconstruction, this method only considers gradient information in horizontal and vertical directions. Compared with the method in [2], the algorithm in [32] extracts gradient information in  
70 more directions. This algorithm uses the Robinson compass operator as the derivative filter, and it rotates the single kernel in eight directions. The eight newly generated kernels are used to extract the input LR images gradient information in eight directions. The extracted information is used to predict the missing high-frequency details. In this method, more directional information improves the reconstructed accuracy.

75 The above two methods address the single-image SR problem by a global mapping model. The literature [6] starts with the initial idea of the literature [2], and it trains multiple mapping models to reconstruct the HR image. This algorithm computes directional features based on all the pixel positions in an image patch. Then, the dimensionality of directional features is decreased by PCA. In the training phase, this  
80 method first clusters the training data by K-means. For each cluster, a mapping model is calculated by ELM, and it is a non-linear mapping model from the LR space to HR space. In the testing phase, the cluster centroid with the minimum distance to the LR image patch is first found. Then the corresponding mapping model is used to generate the HR image patch. All of the generated HR image patches form the target HR output.

85 The three ELM-based SR methods [2] [32] [6] learn the non-linear mapping models from the LR space to HR space, by which the reconstructed HR is generated. Recent years, some regression-based single-image SR methods have also achieved good results.

Timofteet et al. proposed ANR method [34] and A+ method [35], and the A+  
90 method is the improved version of the ANR method. Both of them adopt the piecewise linearization scheme. Timofteet et al. assumed that the mapping between the LR space and HR space is locally linear. Thus, many linear regressors are learned and anchored to the feature space. The ANR and A+ methods use the sparse dictionaries in a different way, whereby the process of computing encoding coefficients is taken place by  
95 choosing the anchored point. This change increases the running speed and improves the reconstruction accuracy. Tang et al. employed the sparse coding and the struc-

tured output regression machine (SORM) to learn the non-linear mapping relationship between the LR feature space to corresponding HR feature space, whereby a global non-linear optimization model is adopted to improve the accuracy.

100 The SR methods in [16] [15] [17] [30] [39] are decision-tree-based, whereby the random forest is comprised of multiple decision trees. All the methods in [16][15][17][30] construct an SR decision tree. Each leaf node of the SR decision tree is corresponding to a linear mapping model from the LR space to HR space. The method in [16] uses all the training data to initialize the root node of the decision tree, then it progressively constructs an SR decision tree with a hierarchical structure. The methods in [15][17][30] 105 first divide the training data into many subsets, each of which is used to construct an SR decision tree. All the SR decision trees form the SR random forest. Both the generated SR decision tree and the SR random forest represents a piecewise linear SR system respectively. Xiong et al. learned the mapping model by a multi-output gradient boosting framework, whereby the decision tree is used as a basic learner. In this method, the 110 learning process of the SR decision tree is guided by gradient boosting.

Methods in [8][22][33] are all based on convolutional neural network. [8] has a simple network structure of three layers, whose training time approximates to three days. The method in [33] employs the residual learning to reduce the difficulty of 115 training a deep network. When the depth of the network increases, this method uses the recursive learning to control the number of the parameters of network. The feature extraction branch consists of many convolutional layers and a transposed convolutional layers, which upsamples the extracted features by a upscaling factor of two. In the reconstructing phase, the input LR image is first upsampled, then the upscaled image 120 is added to the predicted residual image to generate the HR image. The LR image is progressively upsampled, whereby the traditional bicubic interpolation algorithm is replaced by a transposed convolutional layer.

The convolutional-neural-network-based SR methods have greatly improved the reconstruction accuracy. They can estimate the complicated non-linear mapping directly 125 from the input samples. However, the training time is too long. In addition, all the parameters are tuned during the training process. The popular learning algorithm for convolutional neural network is back propagation. It can effectively calculate the gra-

dient by passing the output forward to the input. However, there are some problems. The smaller learning rate will leads to slower convergence speed, and the bigger learning rate will result in instability and divergence. Moreover, the back propagation may convergence to a local minimum. Overtraining can also leads to poor generalization performance.

Compared with back propagation, ELMs training speed is much faster. The proposed single-image SR method learns the non-linear mapping relationship between the LR space and HR space using ELM. Experimental results show that the reconstructed images by our SR method are much closer to the original HR images, and the proposed method has a much faster running speed.

### 3. Extreme Learning Machine (ELM)

Extreme learning machine (ELM) is a fast learning algorithm for training a single-hidden-layer feedforward neural networks (SLFNs) [13][14][12]. One of its advantages is that we don not need to tune the parameters of hidden layer. Moreover, the weights of the output layer can be calculated through a closed form solution. For a generalized single-hidden-layer feed-forward neural network, the output function of one node is formulated as

$$f(x) = \sum_{i=1}^L \beta_i h_i(x) = h(x)\beta \quad (2)$$

where  $L$  is the number of neurons,  $\beta = [\beta_1, \dots, \beta_L]^T$  is the output weights between the hidden layer and the output nodes, and  $h(x) = [h_1(x), \dots, h_L(x)]$  represents the output of the hidden layer. The goal of ELM is not only the minimum training error, but also the minimum norm of output weights. According to the introduction in[4], the smaller the norm of the output weights, the better the generalization ability of the network. ELM needs to minimize the following function:

$$\arg \min_{\beta} ||H\beta - Y_t||^2 + C||\beta||^2 \quad (3)$$

Where  $Y_t$  is the matrix represents the target,  $C$  is a regularization parameter, and  $H$  denotes the matrix of output. The formula of  $H$  is

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \vdots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \quad (4)$$

When the number of training samples is bigger than the number of hidden nodes, the solution of formula Eq. (4) can be described as:

$$\beta = (\frac{I}{C} + H^T H)^{-1} H^T Y_t \quad (5)$$

When we obtain  $\beta$  by formula Eq. (5), the output function of ELMs regressor is

$$f(x) = h(x)\beta = h(x)(\frac{I}{C} + H^T H)^{-1} H^T Y_t \quad (6)$$

To obtain the result of formula Eq. (6), we must compute the output of the hidden layer:

$$H(x) = [G(a_1, b_1, x), \cdots, G(a_L, b_L, x)] \quad (7)$$

Where  $a_i$  represents the weights connect the input layer and the hidden layer,  $b_i$  is the bias of the hidden layer, and  $G()$  denotes the activation function of the hidden layer. The input weights and bias of the hidden layer can be randomly generated based on any continuous probability distribution. The activation function can be any piecewise non-linear continuous function that satisfies the universal approximation capability theorem of ELM [1]. ELM is not sensitive to the number of the hidden nodes. Moreover, the learning speed of ELM is very fast. These advantages make ELM user friendly and efficient. Fig. 1 can give the main idea of solving ELMs.

#### 4. Proposed Method

The core idea of the proposed method is that the feature space is divided into many subspaces, and a certain mapping model is learned for each of the subspace. At this

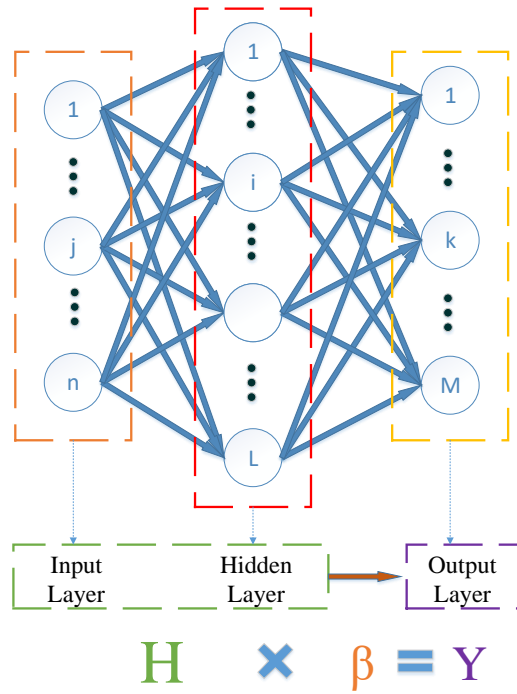


Figure 1: The main idea of solving ELMs. The parameters values can be generated randomly. And  $H$  can be computed immediately. Then we can treat the first two layers as  $H$  and ELM changes to a simple linear regression form. The parameters  $\beta$  between hidden layer and the output layer can be easily and quickly solved.



point, all the mapping models form the piecewise non-linear SR system. The proposed method mainly includes two parts: the training phase for piecewise non-linear mapping models and the SR reconstruction phase.

#### 4.1. Piecewise non-linear mapping model

We extract LR-HR image patch pairs as the training data. The proposed method employs a coarse-to-fine strategy to classify training data. When the partition is complete, for each subspace, we use the training data arriving here to learn a mapping model. We divide the training data into many subspaces, simultaneously an SR decision tree is constructed. The number of the subspaces equals to the number of leaf nodes. Each leaf node is corresponding to a mapping model. In the SR reconstruction phase, the proposed method searches the appropriate mapping model for each input LR image patch. Fig. 2 is the flowchart of the training phase.

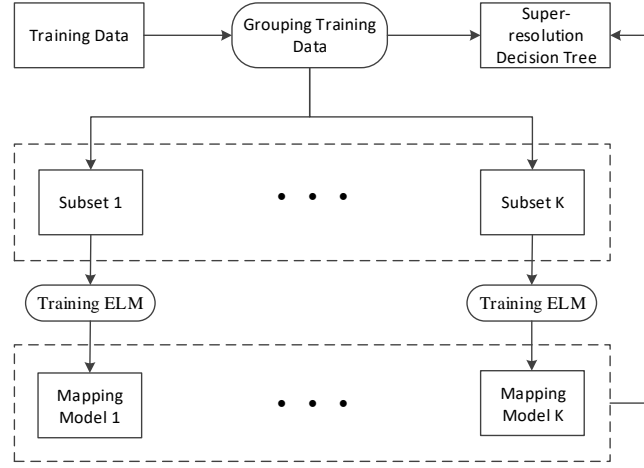


Figure 2: Flowchart outlining the training phase

The selection of image features and algorithm of the decision tree will affect the running speed and the SR reconstruction result. Luo et al [27] proposed method performs the Hadamard transform on LR image patches to extract their image features (Hadamard pattern). The Hadamard transform only involves in addition and subtraction, which makes the transformation simple and fast. Then, the algorithm select two

threshold values and divide patches into three parts, according to the results of the generated Hadamard patterns. This is the rule of dividing tree nodes. Each leaf node of the decision is corresponding to a set of training data. Then, we compute the mapping  
170 model using ELM for each leaf node. Non-leaf nodes in the SR decision tree stores the threshold values and the indices that point to its child nodes. Each leaf node stores the index that points to the corresponding mapping model. Small LR image patches contain little information, while large ones cause much more parameters and over-fitting. The optional side length of the LR image patches is  $2 \times 2$ ,  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ , and  
175  $32 \times 32$  and so on. The order of the corresponding Hadamard matrix is 4, 16, 64, 256, and 1024 and so on. Thus, we set the size of the LR image patch to 44, and we adopt a Hadamard matrix of 16-order as [27]. The first column of  $Q_{16}$  is all ones, and we delete it. The reduced Hadamard matrix is  $Q_{15}$ . Later, the so called Hadamard matrix is  $Q_{15}$ .

Each mapping model consists of three parts as mentioned above: the randomly generated input coefficient matrix and hidden layer bias, and the output coefficient matrix that is calculated mathematically. We use ELM to compute the mapping model. The LR training data are the inputs of the single-hidden-layer feed-forward neural network and the HR training data are the targets. First, a randomly generated coefficient matrix maps the LR training data to the hidden layer. Then, the randomly mapped results are add the randomly generated hidden layer bias. The sum is processed by the activation function, and it is the output of the hidden layer. Last, the output coefficient matrix is calculated using the output of the hidden layer and the target values. For the leaf node  $q$ ,  $H_q$  represents its HR training data, and  $L_q$  denotes its LR training data.  $L_q$  and  $H_q$  are matrix that are stacked by multiple vectorized LR patches and HR patches respectively. The calculation of the mapping model are formulated as

$$H^* = w_q L_q + b_q \quad (8)$$

$$H = G(H^*) \quad (9)$$

180 Where  $w_q$  is the randomly generated input coefficient matrix,  $b_q$  represents the randomly generated hidden layer bias.  $H^*$  denotes the input of the hidden layer.  $G(\bullet)$  is the activation function, and a sigmoid function is adopted in our method.  $H$  represents the output matrix of the hidden layer

$\beta_q$  is the output coefficient matrix that connects the hidden layer and the output  
 185 layer. We can get it by Eq. 5. For the leaf node  $q$ , its mapping models consists of  $w_q$ ,  $b_q$ , and  $\beta_q$ . The proposed method constructs an SR decision tree. Each leaf node is corresponding to a non-linear mapping model that describes the mapping relationship between the LR space to HR space. Our method addresses the single-image SR problem at patch level. For each LR image patch, there is only one mapping model  
 190 corresponds to it. We dont need to preprocess the input LR image using basic interpolation algorithm. It effectively reduces computations and the smoothness in the output HR image.

#### 4.2. SR reconstruction scheme

We first extract LR image patches from the input LR image, then, we perform  
 195 the Hadamard transform on them to compute their Hadamard patterns. The generated Hadamard patterns are the bases of searching their corresponding mapping models. Their Hadamard patterns are compared with the threshold values in the SR decision tree, then, these LR image patches are passed to their corresponding leaf nodes and mapped to the HR space. All the predicted HR patches form the target output. In the  
 200 searching process of mapping models, a ternary search algorithm is employed. All the generated HR patches form the output HR image. Fig. 3 is flowchart of the SR reconstruction.

### 5. Experimental Results

The training dataset in our method contains 91 natural images [42] . We use [27]  
 205 as our decision tree ,which employs a ternary searching algorithm. The size of the LR training image patch is set to 44, and the size of the corresponding HR training image patch is set to  $ss$  ( $s$  is the upscaling factor). In the SR reconstruction phase, when the

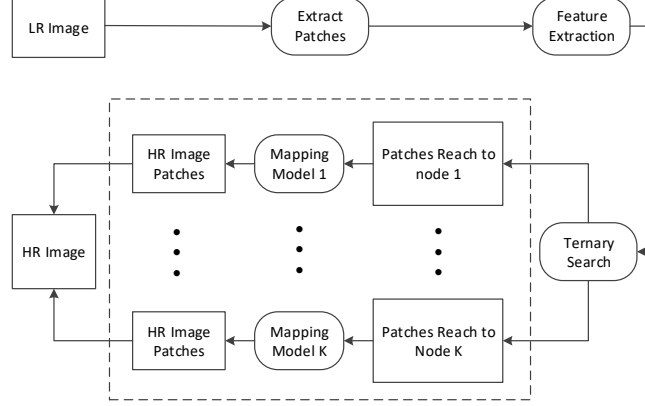


Figure 3: Flowchart outlining the SR reconstruction phase

upsampling factor is 2, the size of the generated HR image patch is 22. No matter in the training phase or the reconstruction phase, the extracted LR image patches have overlapping regions, and the generated HR image patches have no overlapping regions as [27]. The minimum size of each leaf nodes training samples is set to 512, and the parameter  $v$  of the decision tree is set to 0.5, which is same to [27]. In the proposed method, the YCrCb color space is used. The SR reconstruction is only performed on luminance channel. Human is more sensitive to high-frequency brightness changes than high-frequency color changes. Thus, many other SR methods employ the same settings. To generate RGB color images, the proposed method first changes the RGB color space to YCrCb color space, then it performs the SR reconstruction on the luminance channel. The chromatic channels are upscaled by the bicubic interpolation algorithm. Last, our method changes the YCrCb color space to RGB color space to generate the HR output.

### 5.1. Experiment on images blurred by Gaussian filter

To verify the effectiveness of the proposed method, we compared it with other SR methods in image quality (i.e., PSNR and SSIM) and running time. The ANR method[34], the A+ method [35], the SRHDT\_f\* method [16], the RFL method [30], the SRCNN method [8], and the LapSRN method [22] are used for comparison. We

use their published implementations. The upscaling factor is set to 2. We use the configuration recommended by the respective author. Our computing platform was an Intel Core i7-4790 CPU 3.60 GHZ 4 core processor with 16GB RAM.

#### 5.1.1. Experimental Setting

230 In this experiment, to imitate the real imaging process, all the images in testing image datasets Set5 and Set14 were first blurred by a Gaussian filter and then down-sampled by bicubic interpolation with a scaling factor of two to generate the corresponding LR testing images. The size of the Gaussian filter is  $5 \times 5$ , and the standard deviation is 1.6. The pre-trained mapping models, testing images, and the testing codes  
235 can be downloaded at this URL: <https://github.com/youyouyimu/ELMbSISR>.

#### 5.1.2. The Number of Hidden Nodes

In the proposed method, the number of hidden nodes is have a great effect on SR reconstruction performance. ELM is not sensitive to the number of hidden nodes [12]. However, an appropriate number of hidden nodes can achieve better accuracy. Our  
240 method is intended to reconstruct the HR image with high quality from the LR image. To determine the optimal number of hidden nodes, we have conducted an experiment. The results are shown in Fig. 4. Obviously, when the number of hidden nodes increases from 10 to 30, the improvement of the average PSNR on Set5 is 1.1dB. When the number of hidden nodes is bigger than 40, the average PSNR on Set5 gradually decreases.  
245 Moreover, the bigger the number of hidden nodes, the longer the training time. We have to compute more parameters. Thus, in the proposed method, we set the number of hidden nodes to 35.

#### 5.1.3. Experimental Analysis

As can be observed in Fig. 5 and Table1, in comparison with the ANR method[34],  
250 the A+ method[35] [31], the SRHDT\_f\* method[16] [46], the RFL method[30], the SRCNN method[8], the LapSRN method[22], and PLHT method(Piecewise linear regression-based single image super-resolution via Hadamard transform)[27], our proposed method achieves the highest average PSNR with the fast running speed. Our

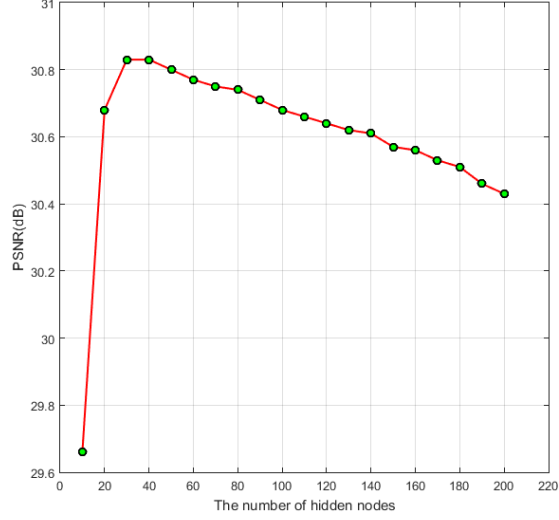


Figure 4: Upscaling factor is two. The relationship between the average PSNR on Set5 and the number of hidden nodes.

method better balances the reconstruction accuracy and the running time. The upscaling factor is two. Figure 6 shows the normalized average PSNR, SSIM, and running time on Set5. Table 2, 3, and 4 shows the PSNR, SSIM, and running time by different methods on Set14.

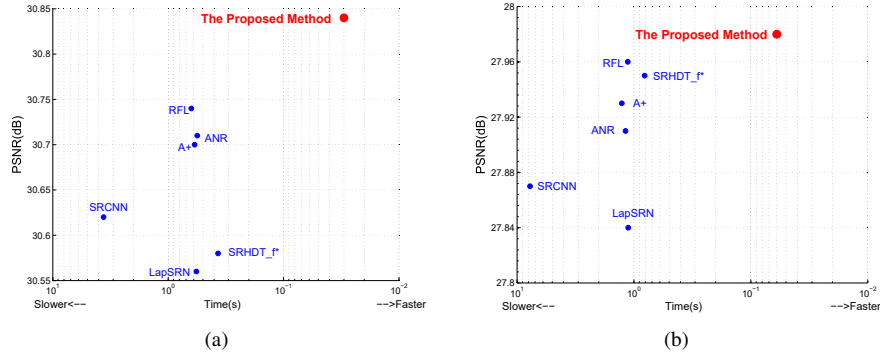


Figure 5: Trade-off between the accuracy and the running speed. The upscaling factor is two. (a) and (b) show the trade-off between the average PSNR and the running time. The proposed method is the best.

We divide the comparison methods into three parts: sparse-coding based methods (ANR and A+), decision tree based methods (SRHDT\_f\*, RFL and PLHT) and con-

Table 1: Upscaling factor is two. The average PSNR, SSIM, and running time obtained by different methods on Set5 and Set14. The **red** font indicates that the corresponding item ranks first, and the **blue** font indicates that the corresponding item ranks second. The proposed method balances the accuracy and the running time better.

Dataset	Measures	Methods								
		Bicubic	ANR	A+	SRHDT_f*	RFL	SRCNN	LapSRN	PLHT	OUR
Set5	PSNR	29.96	30.71	30.7	30.58	<b>30.74</b>	30.62	30.56	<b>30.89</b>	<b>30.89</b>
	SSIM	0.8615	0.8797	0.8810	0.8793	0.8805	0.8783	0.8804	<b>0.8825</b>	<b>0.8819</b>
	TIME(s)	0	0.56	0.59	0.37	0.63	3.63	0.57	<b>0.02</b>	<b>0.03</b>
Set14	PSNR	27.3	27.91	27.93	27.95	27.96	27.87	27.84	<b>27.99</b>	<b>28.02</b>
	SSIM	0.7673	0.7913	<b>0.793</b>	0.79	<b>0.7919</b>	0.7909	<b>0.793</b>	0.7913	0.7903
	TIME(s)	0	1.18	1.27	0.81	1.13	7.74	1.12	<b>0.03</b>	<b>0.06</b>

Table 2: Upscaling factor is two. The PSNR obtained by different methods on Set14. **Bold** font indicates that the corresponding item ranks first.

Images	Methods							
	Bicubic	ANR	A+	SRHDT f*	RFL	SRCNN	LapSRN	OUR
Baboon	23.17	23.47	<b>23.48</b>	<b>23.48</b>	23.47	<b>23.48</b>	<b>23.48</b>	23.44
Barbara	26.11	26.46	<b>26.47</b>	<b>26.47</b>	26.45	26.46	26.44	26.4
Bridge	24.27	24.74	24.77	<b>24.79</b>	24.78	24.77	24.74	<b>24.79</b>
Coastguard	26.49	27.04	<b>27.08</b>	27.04	27.05	27.04	27.06	<b>27.08</b>
Comic	23	23.59	23.6	<b>23.67</b>	23.59	23.57	23.56	23.50
Face	32.65	33.1	<b>33.11</b>	33.02	33.04	33.06	<b>33.11</b>	32.99
Flowers	27.04	27.71	27.73	<b>27.78</b>	27.7	27.68	27.63	27.67
Foreman	30.25	31.3	31.33	31.38	31.59	31.17	31.21	<b>32.26</b>
Lenna	31.33	32.01	32	31.98	31.99	31.92	31.92	<b>32.02</b>
Man	26.87	27.38	27.4	<b>27.41</b>	27.39	27.35	27.32	27.37
Monarch	29.07	29.87	29.9	29.97	29.95	29.81	29.71	<b>30.03</b>
Pepper	31.95	32.66	32.7	32.7	32.79	32.54	32.53	<b>32.91</b>
PPT3	23.64	24.36	24.43	24.52	24.51	24.32	24.2	<b>24.55</b>
Zebra	26.15	27.07	27.06	27.08	27.09	26.96	26.89	<b>27.19</b>
Average	27.3	27.91	27.93	27.95	27.96	27.87	27.84	<b>28.02</b>

260 volutional neural network based methods(SRCNN and LapSRN). Firstly, The sparse-  
coding based methods always need a long time to find the solution. Secondly, The  
SRHDT\_f\* method is a decision-tree-based SR method, and the RFL method is random-  
forest based. The RFL method constructs a SR random forest, and the number of the  
decision tree is much more than the proposed method. Both the SRHDT\_f\*method and  
265 RFL method have to spend more time to searching the mapping models. PLHT use  
same a decision tree as ours,but linear functions in PLHT loss some abilities in map-  
ping. The comparison results show that the proposed method obtains a better trade-off  
between the reconstruction accuracy and the running time.

Table 3: Upscaling factor is two. The SSIM obtained by different methods on Set14. **Bold** font indicates that the corresponding item ranks first.

Images	Methods							
	Bicubic	ANR	A+	SRHDT_f*	RFL	SRCNN	LapSRN	OUR
Baboon	0.5299	0.5624	0.5628	0.5625	0.5618	0.5626	<b>0.5639</b>	0.5617
Barbara	0.7485	0.7691	<b>0.7698</b>	0.7637	0.7679	0.7687	0.7695	0.7660
Bridge	0.6322	0.6685	0.6695	<b>0.6713</b>	0.6678	0.6700	0.6699	0.6676
Coastguard	0.6180	0.6557	0.6581	0.6398	0.6523	0.6566	<b>0.6596</b>	0.6394
Comic	0.6943	0.7335	<b>0.7356</b>	0.7335	0.7342	0.7314	0.7343	0.7276
Face	0.7920	0.8063	0.8071	0.8033	0.8073	0.8088	<b>0.8115</b>	0.8068
Flowers	0.7994	0.8246	<b>0.8257</b>	0.8221	0.8253	0.8235	0.8252	0.8214
Foreman	0.8921	0.9071	0.9096	0.9127	0.9099	0.9051	0.9093	<b>0.9143</b>
Lenna	0.8531	0.8671	0.8685	0.8639	0.8667	0.8654	<b>0.8686</b>	0.8666
Man	0.7429	0.7671	<b>0.7678</b>	0.7644	0.7661	0.7667	0.7674	0.7655
Monarch	0.9174	0.9294	0.9309	0.9291	0.9298	0.9277	<b>0.9304</b>	<b>0.9309</b>
Pepper	0.8629	0.8741	0.8763	0.8744	0.8764	0.8728	0.8765	<b>0.8775</b>
PPT3	0.8772	0.8994	0.9058	0.9055	<b>0.9066</b>	0.9012	0.9019	0.9059
Zebra	0.7812	0.8140	0.8150	0.8133	<b>0.8151</b>	0.8125	0.8146	0.8136
Average	0.7673	0.7913	<b>0.7930</b>	0.7900	0.7919	0.7909	<b>0.7930</b>	0.7903

Table 4: Upscaling factor is two. The running time obtained by different methods on Set14. **Bold** font indicates that the corresponding item ranks first. The running time of the Bicubic method is the baseline, which is not used for comparison.

Images	Methods							
	Bicubic	ANR	A+	SRHDT_f*	RFL	SRCNN	LapSRN	OUR
Baboon	0	1.11	1.20	1.13	1.24	8.14	1.10	<b>0.07</b>
Barbara	0	1.86	2.03	1.29	1.93	14.53	2.00	<b>0.12</b>
Bridge	0	1.18	1.32	1.38	1.35	9.06	1.25	<b>0.07</b>
Coastguard	0	0.51	0.55	0.37	0.56	2.41	0.51	<b>0.03</b>
Comic	0	0.46	0.49	0.47	0.55	2.43	0.58	<b>0.02</b>
Face	0	0.40	0.42	0.21	0.45	1.97	0.44	<b>0.02</b>
Flowers	0	0.93	1.00	0.76	0.94	6.16	0.91	<b>0.05</b>
Foreman	0	0.52	0.56	0.29	0.54	2.41	0.49	<b>0.03</b>
Lenna	0	1.43	1.52	0.68	1.22	9.16	1.26	<b>0.07</b>
Man	0	1.43	1.53	1.08	1.34	9.03	1.26	<b>0.08</b>
Monarch	0	2.12	2.27	1.07	1.80	13.77	1.86	<b>0.11</b>
Pepper	0	1.43	1.51	0.66	1.24	9.08	1.26	<b>0.07</b>
PPT3	0	1.87	2.00	0.85	1.54	12.49	1.63	<b>0.09</b>
Zebra	0	1.23	1.31	1.12	1.16	7.70	1.11	<b>0.06</b>
Average	0.	1.18	1.27	0.81	1.13	7.74	1.12	<b>0.06</b>



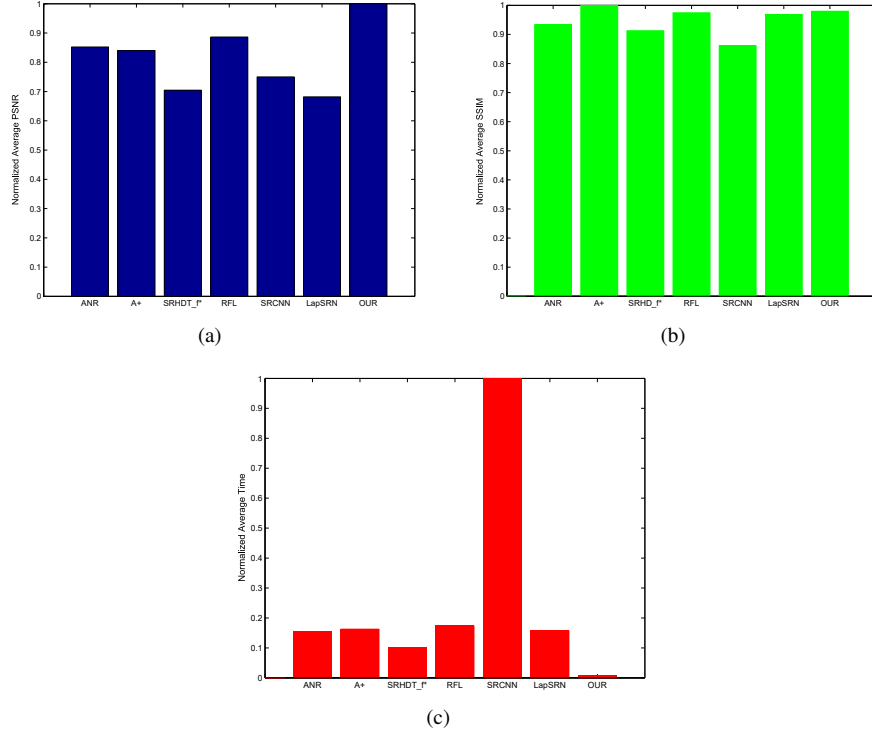


Figure 6: Normalized average PSNR, SSIM and running time by different methods for a scaling factor of two on Set5.

Finally, The SRCNN method and LapSRN method are CNN method. The training dataset of the SRCNN method contains 395909 images, while LapSRN method contains 291 images. However, the proposed method only use 91 natural images for training. As shown in Table. 1, our method achieves the higher average PSNR and SSIM than the SRCNN method on Set5 and Set14. Moreover, our methods running speed is much faster which can be easily found in Figure 6. The proposed method employs the Hadamard patterns and ELM. Based on the generated Hadamard patterns, our method divides the feature space into many non-linear subspaces, and it effectively estimates the whole non-linear feature space. Thus, the proposed method obtains the higher reconstruction accuracy. In addition, the use of ELM greatly reduces the training time.

Table 5: Upscaling factor is two. The average PSNR, SSIM, and running time obtained by different methods on Set5 and Set14. The **red** font indicates that the corresponding item ranks first, and the **blue** font indicates that the corresponding item ranks second. The proposed method balances the accuracy and the running time better.

Dataset	Measures	Methods					
		Bicubic	SME	FIRF	LapSRN	PLHT	OUR
Set5	PSNR	33.7	33.91	33.39	<b>37.44</b>	34.9902	<b>35.09</b>
	SSIM	0.9304	0.9351	0.9323	<b>0.9590</b>	0.9436	<b>0.9437</b>
	TIME(s)	0	74.75	0.53	0.56	<b>0.01</b>	<b>0.03</b>
Set14	PSNR	30.26	30.16	29.15	<b>32.99</b>	31.28	<b>31.33</b>
	SSIM	0.8695	0.8751	0.8674	<b>0.9125</b>	<b>0.8902</b>	0.8892
	TIME(s)	0	161.58	1.26	1.11	<b>0.03</b>	<b>0.07</b>

## 280 5.2. Experiment on images without blurring

To show our methods effectiveness, the Bicubic interpolation algorithm is used as a baseline. The SME method[28], the FIRF method [17], and the LapSRN method [22] were used for comparison. The computing platform was an Intel Core i7-4790 CPU 3.60 GHZ 4 core processor with 16GB RAM. All the testing codes of other methods  
285 were downloaded from the authors websites respectively, and all the parameters are set according to the authors recommendation.

### 5.2.1. Experimental Settings

Different from Experiment above, this experiment generates testing images in a different way. All the images in Set5[7] and Set14[43] are directly downsampled by the  
290 bicubic interpolation algorithm, and the upscaling factor is 2. The number of hidden nodes is set to 35. The rest of parameters are same to the last experiment. The pre-trained mapping models, testing images, and the testing codes can be downloaded at this URL: <https://github.com/youyouyimu/ELMbsISR>.

### 5.2.2. Experimental Analysis

295 Table 5 shows the objective evaluation for all methods. We treat the bicubic method as the baseline and set its running time to zero. Subsequent analysis of the running speed is based on the comparison of the proposed method with other SR methods. As can be observed in Table 5, the proposed method is fastest. The proposed method get the second place on average PSNR and SSIM followed by LapSRN. Our method

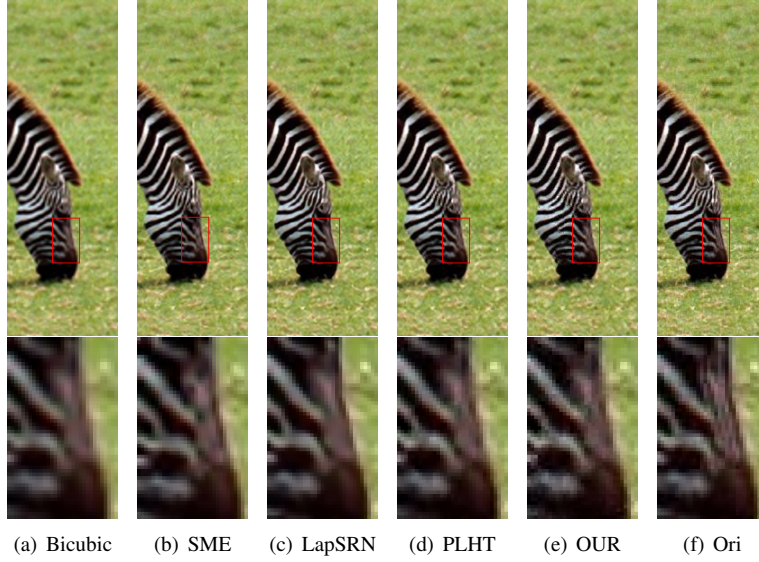


Figure 7: Comparisons between the Bicubic method, SME method, LapSRN method, and the proposed method. Ori is the original image. The proposed method recover more details, and its reconstructed image is much closer to the original image.

300 balances the SR reconstruction accuracy and the running time better. Based on Set5, the average PSNR of our method is 1.27dB higher than that of the FIRF method, and the average SSIM of our method is 0.0067 higher. Moreover, the running speed of our method is much faster, and the speed-up with regard to the FIRF method is 18. Compared with the SME method, the proposed method still obtains the higher average  
305 PSNR and SSIM. In addition, the speed-up of running speed is 2491.67. Based on Set14, the proposed method is still competitive in the PSNR and SSIM. The running speed of our method is the fastest. The speed-up with respect to the FIRF method and SME method is 18 and 2308.29 respectively.

Figure 7, Figure 8, Figure 9 and Figure 10 shows the visual comparison of all  
310 methods on the image 'zebra' and 'flower'. The FIRF method algorithm only deals with the gray images. Thus, we change the reconstructed images into gray images to perform subjective evaluation, as shown in Figure 8 and Figure 10. We perform close-up on reconstructed details in the red box.

As shown in Figure 7, it is easy to find that the reconstructed results of the Bicu-

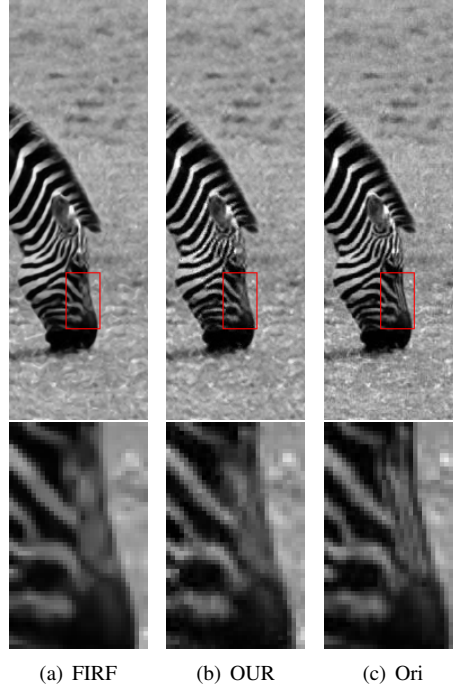


Figure 8: Comparisons between the FIRF method and the proposed method. Ori is the original image. The proposed method recover more details, and its reconstructed image is much closer to the original image.

315 bic method, the SME method, and the LapSRN method are too smooth, especially the  
region below the eyes of this zebra. In the original image, the pixels in white stripes ir-  
regularly changes, and there are differences in brightness between them. The proposed  
method can better reconstruct the texture details in the white stripes, and it can also  
recover more high-frequency details. Thus, the proposed method is much closer to the  
320 original image. In Figure 8, the SR reconstruction result of the FIRF method is too  
smooth, especially the white stripes and the region above the nose. In comparison with  
them, the proposed method can recover more texture details (high-frequency details),  
and its vision quality is better.

Figure 9, and Figure 10 show the reconstructed images of all methods on image  
325 'flower'. The proposed methods reconstructed image shows that our method can better  
recover the texture details between the red dots on the petals. As we can see from  
the original image, the region between the red dots on the petals is not pure white.

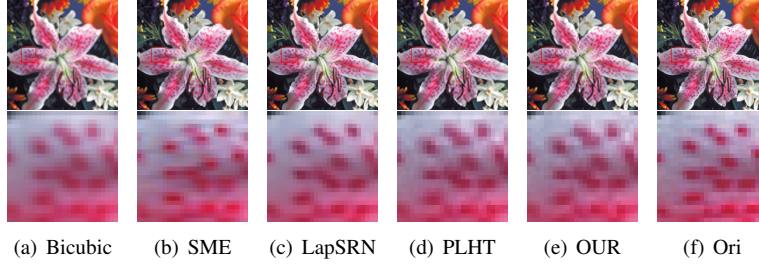


Figure 9: Comparisons between the Bicubic, SME, LapSRN, PLHT methods and the proposed method. Ori is the original image. The proposed method recovers more details, and its reconstructed image is much closer to the original image.

The texture in the region is complicated. The bicubic method, SME method, LapSRN method, and FIRF method smooth this kind of region, and the original texture does not exist. The proposed method can recover texture details well.

Objectively, the proposed method obtains better evaluation results than the bicubic method, SME method, and FIRF method. Subjectively, our method performs better in recovering high-frequency details, and our reconstructed images are much closer to the original images. The LapSRN method performs best in the objective evaluation. We only test its running time on a CPU platform, and our method achieves a faster running speed. The speed-up on Set5 is 18.67, and that on Set14 is 15.86. The reconstructed images of the LapSRN method involve heavy smoothness. However, the proposed methods SR reconstructed images contain much more high-frequency details. The

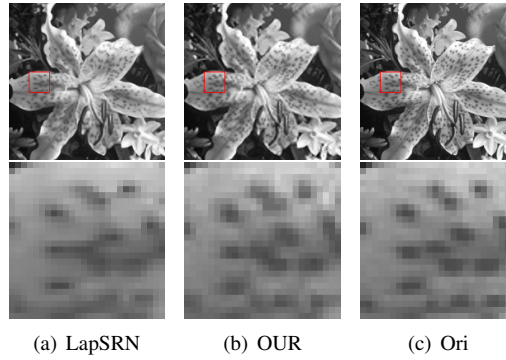


Figure 10: Comparisons between the FIRF methods and the proposed method. Ori is the original image. The proposed method recovers more details, and its reconstructed image is much closer to the original image.

SME method is sparse-coding based, and the FIRF method is interpolation based. The  
340 SME method must mix the outputs that are from many directional interpolators to  
generate the target output. The FIRF method consists of two phases. Each phase it  
must generate lots of SR decision trees. Furthermore, it has to fuse all the predicted  
values to generate the target value. Thus, both the SME method and FIRF method  
have a slower running speed in the reconstruction phase. The proposed method only  
345 constructs one SR decision tree, and it obtains the target value through one prediction.  
Thus, our method can achieve higher reconstruction accuracy with a faster running  
speed. The operation of fusion results in the smoothness in the reconstructed images  
of the SME method and FIRF method. There is no operation of fusion in our method.  
Thus, our reconstructed images are much clearer.

350 The LapSRN methods network structure is too complicated. It trains a global non-  
linear mapping model, which describes the mapping relationship between the LR space  
and HR space. The deep complicated network makes a longer running time. The Lap-  
SRN method first extracts high-frequency information from the input LR image. Then,  
it upsamples the extracted high-frequency information, simultaneously the input LR  
355 image is upsampled. The sum of the two parts is the target output. The LapSRN meth-  
ods upsamples the extracted high-frequency feature and the input LR image using a bi-  
linear interpolation algorithm. The operation of interpolation makes the reconstructed  
images by the LapSRN method look much smoother. The proposed method doesn't  
involve any operation of interpolation in both the training phase and the reconstruction  
360 phase. Thus, the proposed method can reconstruct clearer images.

## 6. Conclusion

ELM is fast learning algorithm for single-hidden-layer feedforward neural network.  
We propose a ELM-based single-image SR method. The proposed method performs  
the Hadamard transform on LR image patches to extract their Hadamard patterns (im-  
365 age feature). Our method classifies the training data based on the extracted Hadamard  
patterns. When the classification is complete, we learn a non-linear mapping model  
using ELM for each class. All the mapping models form a piecewise non-linear SR

system. While classifying training data, an SR decision tree is constructed. Each leaf node corresponds to a mapping model. Experimental results show that the proposed  
370 method can achieve comparable reconstruction accuracy with a faster running speed.

## 7. Acknowledgments

This work was partly supported by the National Science Foundation of China Grant 61300135, Pearl River Technology Nova Project Grant 201710010020, the Fundamental Research Funds for the Central Universities Grant x2rjD2153900, Hong Kong  
375 Scholars Program Grant XJ2014058, and the Open Project Program of the State Key Lab of CAD&CG Grant A1619.

The authors would like to thank the editors and anonymous reviewers for their comments and suggestions.

## References

- 380 [1] , 2006. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks* 17, 879–892. doi:10.1109/TNN.2006.875977.
- [2] An, L., Bhanu, B., 2012. Image super-resolution by extreme learning machine, in: 2012 19th IEEE International Conference on Image Processing, pp. 2209–2212.  
385 doi:10.1109/ICIP.2012.6467333.
- [3] Baker, S., Kanade, T., 2002. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 1167–1183. doi:10.1109/TPAMI.2002.1033210.
- 390 [4] Bartlett, P.L., 1998. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory* 44, 525–536. doi:10.1109/18.661502.

- [5] Chan, T.F., Ng, M.K., Yau, A.C., Yip, A.M., 2007. Superresolution image reconstruction using fast inpainting algorithms. *Applied and Computational Harmonic Analysis* 23, 3 – 24. URL: <http://www.sciencedirect.com/science/article/pii/S1063520307000218>, doi:<https://doi.org/10.1016/j.acha.2006.09.005>. special Issue on Mathematical Imaging.
- [6] Cosmo, D.L., Inaba, F.K., Salles, E.O.T., 2017. Single image super-resolution using multiple extreme learning machine regressors, in: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 397–404. doi:10.1109/SIBGRAPI.2017.59.
- [7] Dong, C., Loy, C.C., He, K., Tang, X., 2014. Learning a deep convolutional network for image super-resolution, in: European Conference on Computer Vision, Springer. pp. 184–199.
- [8] Dong, C., Loy, C.C., He, K., Tang, X., 2016. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 295–307. doi:10.1109/TPAMI.2015.2439281.
- [9] Dong, W., Zhang, L., Lukac, R., Shi, G., 2013. Sparse representation based image interpolation with nonlocal autoregressive modeling. *IEEE Transactions on Image Processing* 22, 1382–1394. doi:10.1109/TIP.2012.2231086.
- [10] Freeman, W.T., Pasztor, E.C., Carmichael, O.T., 2000. Learning low-level vision. *International Journal of Computer Vision* 40, 25–47. URL: <https://doi.org/10.1023/A:1026501619075>, doi:10.1023/A:1026501619075.
- [11] He, H., Siu, W.C., 2011. Single image super-resolution using gaussian process regression, in: CVPR 2011, pp. 449–456. doi:10.1109/CVPR.2011.5995713.
- [12] Huang, G.B., Zhou, H., Ding, X., Zhang, R., 2012. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and*



- 420 Cybernetics, Part B (Cybernetics) 42, 513–529. doi:10.1109/TSMCB.2011.2168604.
- [13] Huang, G.B., Zhu, Q.Y., Siew, C.K., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks, in: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), pp. 985–990  
425 vol.2. doi:10.1109/IJCNN.2004.1380068.
- [14] Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: Theory and applications. Neurocomputing 70, 489 – 501.  
URL: <http://www.sciencedirect.com/science/article/pii/S0925231206000385>, doi:<https://doi.org/10.1016/j.neucom.2005.12.126>.  
430 neural Networks.
- [15] Huang, J.J., Siu, W.C., 2015. Practical application of random forests for super-resolution imaging, in: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2161–2164. doi:10.1109/ISCAS.2015.7169108.
- [16] Huang, J.J., Siu, W.C., 2017. Learning hierarchical decision trees for single-  
435 image super-resolution. IEEE Transactions on Circuits and Systems for Video Technology 27, 937–950.
- [17] Huang, J.J., Siu, W.C., Liu, T.R., 2015. Fast image interpolation via random forests. IEEE Transactions on Image Processing 24, 3232–3245. doi:10.1109/TIP.2015.2440751.
- 440 [18] Irani, M., Peleg, S., 1991. Improving resolution by image registration. CVGIP: Graphical Models and Image Processing 53, 231 – 239.  
URL: <http://www.sciencedirect.com/science/article/pii/104996529190045L>, doi:[https://doi.org/10.1016/1049-9652\(91\)90045-L](https://doi.org/10.1016/1049-9652(91)90045-L).
- 445 [19] Kim, J., Lee, J.K., Lee, K.M., 2016a. Accurate image super-resolution using very deep convolutional networks, in: 2016 IEEE Conference on Computer Vision

and Pattern Recognition (CVPR), pp. 1646–1654. doi:10.1109/CVPR.2016.182.

- [20] Kim, J., Lee, J.K., Lee, K.M., 2016b. Deeply-recursive convolutional network for  
450 image super-resolution, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1637–1645. doi:10.1109/CVPR.2016.181.
- [21] Kim, K.I., Kwon, Y., 2010. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 1127–1133. doi:10.1109/TPAMI.2010.25.
- 455 [22] Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H., 2017. Deep laplacian pyramid networks for fast and accurate super-resolution, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5835–5843. doi:10.1109/CVPR.2017.618.
- [23] Li, J., Qu, Y., Li, C., Xie, Y., Wu, Y., Fan, J., 2015. Learning local gaussian process regression for image super-resolution. *Neurocomputing* 154, 284 – 295.  
460 URL: <http://www.sciencedirect.com/science/article/pii/S0925231214016580>, doi:<https://doi.org/10.1016/j.neucom.2014.11.064>.
- [24] Lin, Z., Shum, H.Y., 2004. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 83–97. doi:10.1109/TPAMI.2004.1261081.  
465
- [25] Lu, X., Yuan, Y., Yan, P., 2013. Image super-resolution via double sparsity regularized manifold learning. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 2022–2033. doi:10.1109/TCSVT.2013.2244798.
- 470 [26] Lu, X., Yuan, Y., Yan, P., 2014. Alternatively constrained dictionary learning for image superresolution. *IEEE Transactions on Cybernetics* 44, 366–377. doi:10.1109/TCYB.2013.2256347.
- [27] Luo, J., Sun, X., Yiu, M.L., Jin, L., Peng, X., 2018. Piecewise linear regression-based single image super-resolution via hadamard transform. *Infor-*

- 475 mation Sciences 462, 315 – 330. URL: <http://www.sciencedirect.com/science/article/pii/S0020025518304717>, doi:<https://doi.org/10.1016/j.ins.2018.06.030>.
- [28] Mallat, S., Yu, G., 2010. Super-resolution with sparse mixing estimators. *IEEE Transactions on Image Processing* 19, 2889–2900. doi:10.1109/TIP.2010.2049927.
- 480 [29] Ni, K.S., Nguyen, T.Q., 2007. Image superresolution using support vector regression. *IEEE Transactions on Image Processing* 16, 1596–1610. doi:10.1109/TIP.2007.896644.
- [30] Schulter, S., Leistner, C., Bischof, H., 2015. Fast and accurate image upscaling with super-resolution forests, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3791–3799. doi:10.1109/CVPR.2015.7299003.
- 485 [31] Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z., 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883. doi:10.1109/CVPR.2016.207.
- 490 [32] Sidike, P., Krieger, E., Alom, M.Z., Asari, V.K., Taha, T., 2017. A fast single-image super-resolution via directional edge-guided regularized extreme learning regression. *Signal, Image and Video Processing* 11, 961–968. URL: <https://doi.org/10.1007/s11760-016-1045-8>, doi:10.1007/s11760-016-1045-8.
- 495 [33] Tai, Y., Yang, J., Liu, X., 2017. Image super-resolution via deep recursive residual network, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2790–2798. doi:10.1109/CVPR.2017.298.
- 500 [34] Timofte, R., De, V., Gool, L.V., 2013. Anchored neighborhood regression for

fast example-based super-resolution, in: 2013 IEEE International Conference on Computer Vision, pp. 1920–1927. doi:10.1109/ICCV.2013.241.

[35] Timofte, R., De Smet, V., Van Gool, L., 2014. A+: Adjusted anchored neighborhood regression for fast super-resolution, in: Asian Conference on Computer Vision (ACCV), pp. 111–126. doi:10.1007/978-3-319-16817-3\_8.

[36] Wang, H., Gao, X., Zhang, K., Li, J., 2016. Image super-resolution using non-local gaussian process regression. *Neurocomputing* 194, 95 – 106. URL: <http://www.sciencedirect.com/science/article/pii/S0925231216002332>, doi:<https://doi.org/10.1016/j.neucom.2016.01.073>.

[37] Wang, H., Gao, X., Zhang, K., Li, J., 2017. Single image super-resolution using gaussian process regression with dictionary-based sampling and student-  $t$  likelihood. *IEEE Transactions on Image Processing* 26, 3556–3568. doi:10.1109/TIP.2017.2700725.

[38] Wang, S., Zhang, L., Liang, Y., Pan, Q., 2012. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2216–2223. doi:10.1109/CVPR.2012.6247930.

[39] Xiong, D., Gui, Q., Hou, W., Ding, M., 2018. Gradient boosting for single image super-resolution. *Information Sciences* 454-455, 328 – 343. URL: <http://www.sciencedirect.com/science/article/pii/S0020025516316401>, doi:<https://doi.org/10.1016/j.ins.2018.04.075>.

[40] Yang, C.Y., Yang, M.H., 2013. Fast direct super-resolution by simple functions, in: 2013 IEEE International Conference on Computer Vision, pp. 561–568. doi:10.1109/ICCV.2013.75.

[41] Yang, J., Wang, Z., Lin, Z., Cohen, S., Huang, T., 2012. Coupled dictionary

- training for image super-resolution. IEEE Transactions on Image Processing 21,  
530 3467–3478. doi:10.1109/TIP.2012.2192127.
- [42] Yang, J., Wright, J., Huang, T.S., Ma, Y., 2010. Image super-resolution via sparse representation. IEEE Transactions on Image Processing 19, 2861–2873. doi:10.1109/TIP.2010.2050625.
- [43] Zeyde, R., Elad, M., Protter, M., 2012. On single image scale-up using sparse-representations, in: Proceedings of the 7th International Conference on Curves and Surfaces, Springer-Verlag, Berlin, Heidelberg. pp. 711–730. URL: [https://doi.org/10.1007/978-3-642-27413-8\\_47](https://doi.org/10.1007/978-3-642-27413-8_47),  
535 doi:10.1007/978-3-642-27413-8\_47.
- [44] Zhang, K., Tao, D., Gao, X., Li, X., Xiong, Z., 2015. Learning multiple linear mappings for efficient single image super-resolution. IEEE Transactions on  
540 Image Processing 24, 846–861. doi:10.1109/TIP.2015.2389629.